

```

/*****
*                               LIST OF HEADER FILES                               *
*****/
#include<iostream.h>
#include<fstream.h>
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<stdio.h>
#include<dos.h>
#include<time.h>
#include<string.h>
#include<iomanip.h>
#include<math.h>
#include<process.h>
/*****
*                               GLOBAL VARIABLE                               *
*****/
int cor_x=-1,cor_y=-1;
/*****
*                               FUNCTION PROTOTYPE                               *
*****/
void mouse();
void exit_button();
int DetectMouse(void);
void InitMouse(void);
void ShowMouse(void);
void HideMouse(void);
void WhereMouse(int*,int*);
int ButtClicked(void);
int ButtReleased(void);
int book_display(int);
void ubuntu(int,int);
char mouse_click_for_dialoge_box(int a,int b);
int mouse_click_for_signin(int a,int b);
int mouse_click_for_setting(int a,int b);
int mouse_click_for_readbook(int a,int b);
int mouse_click_for_add_book(int a,int b);
int mouse_click_for_signup(int a,int b);
int mouse_click_for_regular_home(int a,int b);
int mouse_click_for_premium_home(int a,int b);
int mouse_click_for_admin_home(int a,int b);
int mouse_click_for_diply(int a,int b);
void loading(void);
void wrong_password(void);
void signin_sucess(void);
void issue_sucess(void);
void return_sucess(void);
void buy_sucess(void);
void sell_sucess(void);
void setting_button(int,int);
char return_confirm(void);
char sell_confirm(void);
char deletebook_confirm(void);
char deleteuser_confirm(void);
char exit_confirm(void);
void logo(int,int,int,int);
void side_bar();
/*****
*                               START OF MOUSE FUNCTION                               *
*****/
void mouse()
{
    int ch,a,b;
    DetectMouse();
    InitMouse();
    ShowMouse();
    void WhereMouse(int*,int*);
    ButtClicked();
    ButtReleased();
    while(ch!=27)
    {

```

```

        if(kbhit())
            ch=getch();
        WhereMouse(&a,&b);
        if(ButtReleased())
        {
            cor_x=a,cor_y=b;
            HideMouse();
            delay(40);
            goto out;
        }
    }
out:
}
int DetectMouse(void)
{
    union REGS in,out;
    in.x.ax=0;
    int86(0X33,&in,&out);
    if(out.x.ax==0)
        return(0);
    else
        return(1);
}
void InitMouse(void)
{
    union REGS in,out;
    in.x.ax=33;
    int86(0X33,&in,&out);
    return;
}
void HideMouse(void)
{
    union REGS in,out;
    in.x.ax=2;
    int86(0X33,&in,&out);
    return;
}
void ShowMouse(void)
{
    union REGS in,out;
    in.x.ax=1;
    int86(0X33,&in,&out);
    return;
}
void WhereMouse(int *x,int *y)
{
    union REGS in,out;
    in.x.ax=3;
    int86(0X33,&in,&out);
    *x=(int)out.x.cx;
    *y=(int)out.x.dx;
    return;
}
int ButtClicked(void)
{
    union REGS imouse,omouse;
    int mc,mr;
    imouse.x.ax = 3;
    int86(0X33,&imouse,&omouse);
    mc=omouse.x.bx;
    return(mc);
}
int ButtReleased(void)
{
    int br;
    br=ButtClicked();
    do {}
    while(ButtClicked()!=0);
    return br;
}
/*****
*                               END OF MOUSE FUNCTION                               *
*****/

```

```

*****/
/*****
*                               *
*****/
class book
{
public:
    int book_id;
    char book_name[40];
    char author[20];
    float price;
    char content[200];
    int issue_stock;
    int buy_stock;
    void del_book(void);
    void read_book(int);
    book()
    {
        book_id=0;
        issue_stock=buy_stock=5;
        price=0.0;
        strcpy(book_name,"");
        strcpy(author,"");
        strcpy(content,"");
    }
};

void book::read_book(int id)
{
    book bk9;
    char exit_ans;
    if(id==0)
    {
        goto end_read_book;
    }
    else
    {
        ifstream filbookread;
        filbookread.open("book",ios::in|ios::binary);
        if(!filbookread)
        {
            cout<<"cannot open file!!";
            delay(2000);
            return ;
        }
        filbookread.seekg(0);
        filbookread.read((char *) &bk9,sizeof(bk9));
        while(filbookread!=0)
        {
            if(id==bk9.book_id)
            {
book_read:
                cleardevice();
                int i,exitid=-1;
                setfillstyle(SOLID_FILL,2);
                bar(15,25,106,157);
                setfillstyle(SOLID_FILL,9);
                bar(10,20,100,150);
                setcolor(15);
                rectangle(0,170,639,479);
                exit_button();
                setcolor(15);
                outtextxy(22,35,"BOOK..");
                outtextxy(150,25,"BOOK NAME:");
                outtextxy(250,25,bk9.book_name);
                outtextxy(150,78,"AUTHOR   :");
                outtextxy(250,78,bk9.author);
                outtextxy(150,131,"PRICE    :");
                gotoxy(32,9);
                cout<<bk9.price;
                gotoxy(2,12);
                cout<<bk9.content;
                mouse();
            }
        }
    }
}

```

```

        exitid=mouse_click_for_readbook(cor_x,cor_y);
        switch(exitid)
        {
            case 1:
                exit_ans=exit_confirm();
                if(exit_ans=='y')
                    return ;
            default:
                goto book_read;
        }
    }
    filbookread.read((char *) &bk9,sizeof(bk9));
}
filbookread.close();
}
end_read_book:
}
class user
{
public:
    int user_id;
    char user_name[20];
    char user_pass[20];
    char contact[20];
    char dob[20];
    int member_type;
    long amount;
    int book_issue_id[3];
    int book_buy_id[50];
    user(void);
    int get_data(void);
    void credits(void);
    void setting(void);
};
user::user()
{
    user_id=0;
    strcpy(user_name,"");
    strcpy(user_pass,"");
    strcpy(contact,"");
    strcpy(dob,"");
    member_type=-1;
    amount=0.0;
    book_issue_id[0]=book_issue_id[1]=book_issue_id[2]=0;
    for(int i=0; i<50; ++i)
    {
        book_buy_id[i]=0;
    }
}
int user::get_data()
{
    char pass[20],member[20];
    strcpy(member,"");
    strcpy(pass,"");
    int x_out=120,y_out=100,i,mouse_signup=-1;
    char exit_choice=NULL;
reenter:
    int check=0;
    clrscr();
    cleardevice();
    setcolor(1);
    settextstyle(0,0,3);
    outtextxy(240,40,"SIGN UP");
    rectangle(100,20,545,450);
    setcolor(5);
    for(i=94; i<=286; i+=32)
    {
        rectangle(275,i,520,i+18);
    }
    setcolor(9);
    settextstyle(0,0,1);
    outtextxy(x_out,y_out,"User_Name:");

```

```

    outtextxy(x_out,(y_out+32),"Password:");
    outtextxy(x_out,(y_out+64),"RE_Enter Password:");
    outtextxy(x_out,(y_out+98),"Contact_NO:");
    outtextxy(x_out,(y_out+130),"Date_Of_Birth:");
    outtextxy(x_out,(y_out+162),"Membership_Type:");
    outtextxy(x_out,(y_out+196),"Enter Amount:");
    setfillstyle(SOLID_FILL,8);
    bar(240,400,400,430); //create account
    setcolor(15);
    outtextxy((x_out+135),(y_out+312),"Create_My_Account");
    exit_button();

signup_mouse:
    mouse();
    mouse_signup=mouse_click_for_signup(cor_x,cor_y); //mouse for sinup
    switch( mouse_signup)
    {
    case 1:
        goto fill_form; //filling all details
    case 2:
        exit_choice=exit_confirm();
        if(exit_choice=='y')
            return -1;
        else if(exit_choice=='n')
            goto reenter;
        break;
    default:
        goto signup_mouse;
    }
fill_form:
    gotoxy(36,7);
    cin.getline(user_name,20);
    gotoxy(36,9);
    cin.getline(user_pass,20);
    gotoxy(36,11);
    cin.getline(pass,20);
    gotoxy(36,13);
    cin.getline(contact,20);
    gotoxy(36,15);
    cin.getline(dob,20);
    gotoxy(36,17);
    cin.getline(member,20);
    gotoxy(36,19);
    cin>>amount;
    cin.ignore();

signup_button:
    mouse();
    int mouse_s=mouse_click_for_signup(cor_x,cor_y);
    switch( mouse_s)
    {
    case 2:
        exit_choice=exit_confirm();
        if(exit_choice=='y')
            return -1;
        else if(exit_choice=='n')
            goto reenter;
    case 3:
        goto check;
    default:
        goto signup_button;
    }
check:
    user us;
    int check1=-1;
    ifstream filinf2;
    filinf2.open("user",ios::in|ios::binary);
    if(!filinf2)
    {
        cout<<"cannot open file!!";
        delay(2000);
        exit(0);
    }
    filinf2.seekg(0);

```

```

while(filinf2)
{
    filinf2.read((char *) &us,sizeof(us));
    if(strcmp(user_name,us.user_name)==0)
    {
        check1=0;
    }
}
filinf2.close();
if(check1==0)
{
    clrscr();
    cleardevice();
    check=0;
    cout<<"user name already used";
    mouse();
    goto reenter;
}
if(strcmp(user_name,"")==0||strcmp(user_pass,"")==0||
    strcmp(dob,"")==0||strcmp(member,"")==0||strcmp(pass,"")
    ==0||strcmp(contact,"")==0)
{
    cout<<"Please fill all details...";
    check=-1;
}
else if(strcmp(user_pass,pass)!=0)
{
    cout<<"Please enter password correctly...";
    check=-1;
}
else if((strcmp(member,"regular")!=0)&&(strcmp(member,"premium")!=0)
    &&(strcmp(member,"admin")!=0))
{
    cout<<"Membership type must be 'regular' or 'premium'...";
    check=-1;
}
else if(amount<5000)
{
    cout<<"Amount must be greater than 5000.";
    check=-1;
}
if(check!=0)
{
    mouse();
    goto reenter;
}
else
{
    if(strcmp(member,"regular")==0)
    {
        member_type=2;
        amount-=500;
    }
    else if(strcmp(member,"premium")==0)
    {
        member_type=1;
        amount-=1000;
    }
    else if(strcmp(member,"admin")==0)
    {
        member_type=0;
        amount-=1000;
    }
}
setcolor(15);
return 0;
}

void user::setting()
{
    int i,ch_setting=-1;
    char exit_choice,member[20];
start_setting:

```

```

clrscr();
cleardevice();
setcolor(3);
settextstyle(3,1,7);
outtextxy(5,150,"SETTING");
setcolor(15);
settextstyle(0,0,1);
setfillstyle(SOLID_FILL,9);
bar(400,0,639,300);
setfillstyle(SOLID_FILL,4);
bar(10,40,90,120);
setting_button(50,78);
gotoxy(15,4);
cout<<"USER NAME      : "<<user_name;
gotoxy(15,7);
cout<<"PASSWORD        : "<<user_pass;
gotoxy(15,10);
cout<<"DATE OF BIRTH   : "<<dob;
gotoxy(15,13);
cout<<"CONTACT NO.     : "<<contact;
switch(member_type)
{
case 0:
gotoxy(15,16);
cout<<"MEMBERSHIP TYPE:ADMINISTRATOR";
break;
case 1:
gotoxy(15,16);
cout<<"MEMBERSHIP TYPE:PREMIUM";
break;
case 2:
gotoxy(15,16);
cout<<"MEMBERSHIP TYPE:REGULAR";
break;
}
gotoxy(15,19);
cout<<"ACCOUNT BALANCE:"<<amount;
for(i=35; i<=290; i+=50)
{
setfillstyle(SOLID_FILL,8);
bar(400,i,639,i+30);
}
exit_button();
outtextxy(420,45,"CHANGE USERNAME");
outtextxy(420,95,"CHANGE PASSWORD");
outtextxy(420,145,"CHANGE DATE OF BIRTH");
outtextxy(420,195,"CHANGE CONTACT NO.");
outtextxy(420,245,"CHANGE MEMBERSHIP TYPE");
outtextxy(420,295,"RECHARGE ACCOUNT");
mouse_pre:
mouse();
ch_setting=mouse_click_for_setting(cor_x,cor_y);
switch(ch_setting)
{
case 1:
gotoxy(10,23);
cout<<"Enter your new user name:";
gotoxy(37,23);
cin>>user_name;
cin.ignore();
break;
case 2:
gotoxy(10,23);
cout<<"Enter your new password:";
gotoxy(35,23);
cin>>user_pass;
cin.ignore();
break;
case 3:
gotoxy(10,23);
cout<<"Enter your new Date of birth:";
gotoxy(41,23);

```

```

        cin>>dob;
        cin.ignore();
        break;
    case 4:
        gotoxy(10,23);
        cout<<"Enter your new contact no.:";
        gotoxy(39,23);
        cin>>contact;
        cin.ignore();
        break;
    case 5:
        gotoxy(10,23);
        cout<<"Enter your new membership type:";
        gotoxy(43,23);
        cin.getline(member,20);
        if(strcmp(member,"regular")==0)
        {
            member_type=2;
        }
        else if(strcmp(member,"premium")==0)
        {
            member_type=1;
        }
        else if(strcmp(member,"admin")==0)
        {
            member_type=0;
        }
        break;
    case 6:
        int amt;
        gotoxy(10,23);
        cout<<"Enter the amount to recharge:";
        gotoxy(41,23);
        cin>>amt;
        amount+=(amt*0.9);
        cin.ignore();
        break;
    case 7:
        exit_choice=exit_confirm();
        if(exit_choice=='y')
            return ;
        else if(exit_choice=='n')
            goto start_setting;
    default:
        goto mouse_pre;
    }
    goto start_setting;
}

void user::credits()
{
    cleardevice();
    int i,j,k,x;
    settextstyle(1,0,2);
    for(x=0; x<350; x++)
    {
        k=random(15)+1;
        setcolor(k);
        i=random(600);
        j=random(460);
        outtextxy(i,j,"HTCL");
    }
    setfillstyle(SOLID_FILL,7);
    setcolor(4);
    settextstyle(0,0,1);
    bar(200,190,420,330);
    outtextxy(275,210,"MADE BY");
    outtextxy(220,240,"Master Dhaval Khatri");
    outtextxy(220,270,"Master Mayank Vaishnav");
    outtextxy(220,300,"Master Deepanshu Sharma");
    mouse();
    setfillstyle(SOLID_FILL,7);
    bar(200,70,420,420);
}

```



```

        setcolor(4);
        outtextxy(260,90,"Special Thanks");
        outtextxy(310,120,"TO");
        setttextstyle(0,0,1);
        outtextxy(240,150,"Mr.Vinod Mishra");
        outtextxy(240,180,"Vatsal Desai");
        outtextxy(240,210,"Sejal Santoki");
        outtextxy(240,240,"Priya Goyani");
        outtextxy(240,270,"Simran Agarwal");
        outtextxy(240,300,"Nikita Patel");
        outtextxy(240,330,"Pryogini Dalai");
        outtextxy(240,360,"Ketan Patel");
        outtextxy(240,390,"Ankit Sudani");
        mouse();
    }
    class admin:public user
    {
    public:
        void delete_user(void);
        void delete_book(void);
        void bookadd(void);
    };
    void admin::bookadd(void)
    {
        book bk;
start:
        int x_out=30,y_out=100,i;
        char exit_choice;
        clrscr();
        cleardevice();
        setcolor(11);
        setttextstyle(0,0,3);
        outtextxy(240,40,"ADD BOOK");
        setcolor(5);
        rectangle(20,25,620,220);
        for(i=94; i<=190; i+=32)
        {
            rectangle(180,i,505,i+18);
        }
        rectangle(0,250,639,430);
        setttextstyle(0,0,1);
        setcolor(11);
        outtextxy(x_out,y_out,"BOOK ID: ");
        outtextxy(x_out,(y_out+32),"BOOK NAME: ");
        outtextxy(x_out,(y_out+64),"AUTHOR NAME: ");
        outtextxy(x_out,(y_out+98),"BOOK PRICE: ");
        outtextxy(x_out,(y_out+130),"BOOK CONTENT: ");
        setfillstyle(SOLID_FILL,4);
        setcolor(0);
        bar(240,440,400,470); //INSERT BOOK
        outtextxy(260,455,"Insert the Book");
        exit_button();
mouse_re:
        mouse();
        int choice=mouse_click_for_add_book(cor_x,cor_y);
        switch(choice)
        {
        case 1:
            goto out;
        case 2:
            exit_choice=exit_confirm();
            if(exit_choice=='y')
                return;
            else if(exit_choice=='n')
                goto start;
        default:
            goto mouse_re;
        }
    }
out:
        gotoxy(24,7);
        cin>>bk.book_id;
        cin.ignore();

```

```

gotoxy(24,9);
cin.getline(bk.book_name,40);
gotoxy(24,11);
cin.getline(bk.author,20);
gotoxy(24,13);
cin>>bk.price;
cin.ignore();
gotoxy(2,17);
cin.getline(bk.content,200);
mouse_ag:
mouse();
int choicel=mouse_click_for_add_book(cor_x,cor_y);
switch(choicel)
{
case 2:
    exit_choice=exit_confirm();
    if(exit_choice=='y')
        return;
    else if(exit_choice=='n')
        goto start;
case 3:
    ofstream filbook3;
    filbook3.open("book",ios::app|ios::binary);
    filbook3.write((char *) &bk,sizeof(bk));
    filbook3.close();
    break;
default:
    goto mouse_ag;
}
goto start;
}
void admin::delete_book()
{
    book bk6,bk7;
re_display:
    int buk_delete_id;
    char confirm='n';
    clrscr();
    cleardevice();
    buk_delete_id=book_display(4);
    if(buk_delete_id==0)
    {
        goto re_display;
    }
    else if(buk_delete_id==-1)
    {
        clrscr();
        cleardevice();
        return;
    }
    ifstream filbooktrans;
    filbooktrans.open("book",ios::in|ios::binary);
    if(!filbooktrans)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    ofstream bookdelete;
    bookdelete.open("temp",ios::out|ios::binary);
    if(!bookdelete)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    filbooktrans.seekg(0);
    filbooktrans.read((char *) &bk7,sizeof(bk7));
    while(filbooktrans)
    {
        if(buk_delete_id==bk7.book_id)
        {

```

```

        confirm=deletebook_confirm();
        if(confirm=='n')
        {
            bookdelete.write((char *) &bk7,sizeof(bk7));
            filbooktrans.close();
            bookdelete.close();
            remove("temp");
            goto re_display;
        }
    }
    else
    {
        if(bk7.book_id>buk_delete_id)
        {
            bk7.book_id-=1;
        }
        bookdelete.write((char *) &bk7,sizeof(bk7));
    }
    filbooktrans.read((char *) &bk7,sizeof(bk7));
}
filbooktrans.close();
bookdelete.close();
remove("book");
rename("temp","book");
goto re_display;
}
void admin::delete_user()
{
    user url,ur2;
start:
    char confirm='n',usr_delete_name[20];
    int dummy=1,last=0,i,delete_id,pos=1,buk_id;
    char exit_choice=NULL;
re_display:
    clrscr();
    cleardevice();
    last=0;
    setcolor(15);
    setfillstyle(SOLID_FILL, BLUE);
    bar(0,0,639,90);
    settextstyle(0,0,4);
    outtextxy(160,30,"DELETE USER");
    settextstyle(0,0,1);
    outtextxy(50,100,"USER NAME");
    outtextxy(250,100,"ACCOUNT TYPE");
    outtextxy(510,100,"ACCOUNT BAL");
    ifstream fileuserdisp;
    fileuserdisp.open("user",ios::in|ios::binary);
    if(! fileuserdisp)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    exit_button();
    if(pos>4)
    {
        setfillstyle(SOLID_FILL,9);
        bar(39,330,130,360);
        outtextxy(55,342,"PREVIOUS");
    }
    setfillstyle(SOLID_FILL,1);
    bar(510,330,590,360);
    outtextxy(535,342,"NEXT");
    fileuserdisp.seekg(0);
    setcolor(15);
    for(int k=0; k<pos; ++k)
    {
        fileuserdisp.read((char *) &url,sizeof(url));
    }
    int d,f=10;
    for(d=148,i=0; i<4&& fileuserdisp!=0; d+=48,++i)

```

```

    {
        rectangle(40,d-11,590,d+21);
        gotoxy(8,f);
        cout<<url.user_name;
        gotoxy(66,f);
        cout<<url.amount;
        gotoxy(42,f);
        cout<<url.member_type;
        f+=3;
        fileuserdisp.read((char *) &url,sizeof(url));
        ++pos;
        last++;
    }
re_ans:
    mouse();
    int choice=mouse_click_for_diply(cor_x,cor_y);
    switch(choice)
    {
    case 0:
    case 1:
    case 2:
    case 3:
        if(choice>last)
        {
            pos-=last;
            goto re_ans;
        }
        delete_id=((pos+choice)-last);
        pos-=last;
        fileuserdisp.close();
        break;
    case 4:
        fileuserdisp.close();
        if(pos>4)
        {
            pos-=(4+last);
        }
        else
            pos=1;
        goto re_display;
    case 5:
        if( fileuserdisp==0)
        {
            pos=1;
            fileuserdisp.close();
        }
        goto re_display;
    case 6:
        exit_choice=exit_confirm();
        fileuserdisp.close();
        setcolor(15);
        if(exit_choice=='y')
            return ;
        else if(exit_choice=='n')
        {
            pos-=last;
            goto re_display;
        }
    default:
        goto re_ans;
    }
    strcpy(usr_delete_name,"");
    ifstream filusertrans;
    filusertrans.open("user",ios::in|ios::binary);
    if(!filusertrans)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    ofstream userdelete;
    userdelete.open("temp",ios::out|ios::binary);

```

```

if(!userdelete)
{
    cout<<"cannot open file!!";
    delay(2000);
    return ;
}
filertrans.seekg(0);
dummy=1;
filertrans.read((char *) &ur2,sizeof(ur2));
while(filertrans)
{
    if(dummy==delete_id)
        strcpy(usr_delete_name,ur2.user_name);
    ++dummy;
    if(strcmp(usr_delete_name,ur2.user_name)==0)
    {
        confirm=deleteuser_confirm();
        if(confirm=='n')
        {
            userdelete.write((char *) &ur2,sizeof(ur2));
            filertrans.close();
            userdelete.close();
            remove("temp");
            goto re_display;
        }
    }
    else
    {
        userdelete.write((char *) &ur2,sizeof(ur2));
    }
    filertrans.read((char *) &ur2,sizeof(ur2));
}
filertrans.close();
userdelete.close();
remove("user");
rename("temp","user");
goto start;
}

class regular:public user
{
public:
    void issue_book(void);
    void return_book(void);
};

void regular::issue_book()
{
    book bk2;
start:
    setcolor(7);
    int repeat=-1,buk_issue_id=-1,buk_id_check=-1;
    int buk_stock_check=-1,check=-1,i,pos,d;
    clrscr();
    cleardevice();
    buk_issue_id=book_display(1);
    if(buk_issue_id==-1)
    {
        clrscr();
        cleardevice();
        return;
    }
    ifstream filbookmatch;
    filbookmatch.open("book",ios::in|ios::binary);
    if(!filbookmatch)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    filbookmatch.seekg(0);
    filbookmatch.read((char *) &bk2,sizeof(bk2));
    while(filbookmatch)
    {

```

```

        if(bk2.book_id==buk_issue_id)
        {
            pos=filbookmatch.tellg();
            buk_id_check=0;
            buk_stock_check=bk2.issue_stock;
            break;
        }
        filbookmatch.read((char *) &bk2,sizeof(bk2));
    }
    filbookmatch.close();
    if(buk_stock_check==0&&buk_id_check==0)
    {
        cleardevice();
        outtextxy(200,350,"Sorry the book is out of stock");
        mouse();
    }
    else if(buk_id_check==0&&buk_stock_check!=0)
    {
        ofstream filbookissue;
        filbookissue.open("book",ios::in|ios::binary);
        if(!filbookissue)
        {
            cout<<"cannot open file!!";
            delay(2000);
            return ;
        }
        for(i=0; i<3; ++i)
        {
            if(book_issue_id[i]==buk_issue_id)
            {
                repeat=0;
                goto out;
            }
            if(book_issue_id[i]==0)
            {
                book_issue_id[i]=buk_issue_id;
                check=0;
                bk2.issue_stock-=1;
                filbookissue.seekp(pos-sizeof(bk2));
                filbookissue.write((char *) &bk2,sizeof(bk2));
                filbookissue.close();
                break;
            }
        }
    }
out:
    cleardevice();
    if(check==0)
        issue_sucess();
    else if(repeat==0)
    {
        outtextxy(200,200,"Book is already issued!!!");
        mouse();
    }
    else
    {
        outtextxy(200,200,"No more books can be issued!!!");
        mouse();
    }
}
goto start;
}
void regular::return_book()
{
    book bk3;
start:
    int i,d,return_id,book_return_index,pos;
    char exit_choice;
return_book:
    int k=0;
    setcolor(7);
    cleardevice();
    setfillstyle(SOLID_FILL, BLUE);

```

```

bar(0,0,639,90);
settextstyle(0,0,4);
outtextxy(160,30,"RETURN BOOK");
settextstyle(0,0,1);
exit_button();
outtextxy(70,100,"BOOK NAME");
outtextxy(350,100,"AUTHOR");
outtextxy(510,100,"PRICE");
ifstream filbookread;
filbookread.open("book",ios::in|ios::binary);
if(!filbookread)
{
    cout<<"cannot open file!!";
    delay(2000);
    return ;
}
filbookread.seekg(0);
filbookread.read((char *) &bk3,sizeof(bk3));
int f=10,dumy[3]= {0,0,0};
setcolor(15);
for(d=148,i=0; i<3;)
{
    if(book_issue_id[i]==bk3.book_id)
    {
        dumy[k]=bk3.book_id;
        rectangle(40,d-11,590,d+21);
        dumy[i]=bk3.book_id;
        gotoxy(8,f);
        cout<<bk3.book_name;
        gotoxy(66,f);
        cout<<bk3.price;
        gotoxy(42,f);
        cout<<bk3.author;
        f+=3;
        d+=48;
        ++i;
        ++k;
        filbookread.seekg(0);
    }
    if(book_issue_id[i]==0)
    {
        break;
    }
    filbookread.read((char *) &bk3,sizeof(bk3));
}
filbookread.close();
return_mouse:
mouse();
int re_id=mouse_click_for_diplay(cor_x,cor_y);
switch(re_id)
{
case 0:
case 1:
case 2:
    if(dumy[re_id]==0)
        goto return_mouse;
    return_id=dumy[re_id];
    break;
case 6:
    exit_choice=exit_confirm();
    if(exit_choice=='y')
        return;
    else if(exit_choice=='n')
        goto return_book;
default:
    goto return_mouse;
}
for(i=0; i<3; ++i)
{
    if(return_id==book_issue_id[i])
    {
        book_return_index=i;
    }
}

```

```

    }
}
ifstream filbookmatch;
filbookmatch.open("book",ios::in|ios::binary);
if(!filbookmatch)
{
    cout<<"cannot open file!!";
    delay(2000);
    return ;
}
filbookmatch.seekg(0);
filbookmatch.read((char *) &bk3,sizeof(bk3));
while(filbookmatch)
{
    if(bk3.book_id==return_id)
    {
        pos=filbookmatch.tellg();
        break;
    }
    filbookmatch.read((char *) &bk3,sizeof(bk3));
}
filbookmatch.close();
ofstream filbookreturn;
filbookreturn.open("book",ios::in|ios::binary);
if(!filbookreturn)
{
    cout<<"cannot open file!!";
    delay(2000);
    return ;
}
book_issue_id[book_return_index]=0;
for(i=book_return_index; i<2; ++i)
{
    book_issue_id[i]=book_issue_id[i+1];
}
book_issue_id[2]=0;
bk3.issue_stock+=1;
filbookreturn.seekp(pos-sizeof(bk3));
filbookreturn.write((char *) &bk3,sizeof(bk3));
return_sucess();
filbookreturn.close();
goto start;
}
class premium:public regular
{
public:
    void buy_book(void);
    void sell_book(void);
};
void premium::buy_book()
{
    book bk4;
start:
    setcolor(7);
    int repeat=-1,buk_buy_id=-1,buk_id_check=-1;
    int buk_stock_check=-1,check=-1,i,pos,d;
    clrscr();
    cleardevice();
    buk_buy_id=book_display(2);
    if(buk_buy_id!=-1)
    {
        clrscr();
        cleardevice();
        return;
    }
    ifstream filbookmatch;
    filbookmatch.open("book",ios::in|ios::binary);
    if(!filbookmatch)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
}

```



```

    }
    filbookmatch.seekg(0);
    filbookmatch.read((char *) &bk4,sizeof(bk4));
    while(filbookmatch)
    {
        if(bk4.book_id==buk_buy_id)
        {
            pos=filbookmatch.tellg();
            buk_id_check=0;
            buk_stock_check=bk4.buy_stock;
            break;
        }
        filbookmatch.read((char *) &bk4,sizeof(bk4));
    }
    filbookmatch.close();
    if(buk_stock_check==0&&buk_id_check==0)
    {
        outtextxy(200,350,"Sorry the book is out of stock");
        mouse();
    }
    else if(buk_id_check==0&&buk_stock_check!=0)
    {
        ofstream filbookbuy;
        filbookbuy.open("book",ios::in|ios::binary);
        if(!filbookbuy)
        {
            cout<<"cannot open file!!";
            delay(2000);
            return ;
        }
        for(i=0; i<3; ++i)
        {
            if(book_buy_id[i]==buk_buy_id)
            {
                repeat=0;
                goto out;
            }
            if(book_buy_id[i]==0)
            {
                book_buy_id[i]=buk_buy_id;
                check=0;
                bk4.buy_stock-=1;
                amount=amount-bk4.price;
                filbookbuy.seekp(pos-typeof(bk4));
                filbookbuy.write((char *) &bk4,sizeof(bk4));
                filbookbuy.close();
                break;
            }
        }
    }
out:
    cleardevice();
    if(check==0)
        buy_sucess();
    else if(repeat==0)
    {
        outtextxy(200,200,"Book is already buyed!!!");
        mouse();
    }
    else
        outtextxy(200,200,"No more books can be bought!!!");
    mouse();
}
goto start;
}
void premium::sell_book()
{
    book bk5;
start:
    int i,d,return_id,book_issue_index,pos;
    char exit_choice;
return_book:
    int k=0;

```

```

setcolor(7);
cleardevice();
setfillstyle(SOLID_FILL, BLUE);
bar(0,0,639,90);
settextstyle(0,0,4);
outtextxy(180,30,"SELL BOOK");
settextstyle(0,0,1);
exit_button();
outtextxy(70,100,"BOOK NAME");
outtextxy(330,100,"AUTHOR");
outtextxy(510,100,"PRICE");
ifstream filbookread;
filbookread.open("book",ios::in|ios::binary);
if(!filbookread)
{
    cout<<"cannot open file!!";
    return ;
}
filbookread.seekg(0);
setcolor(15);
filbookread.read((char *) &bk5,sizeof(bk5));
int f=10,dummy[3]={0,0,0};
for(d=148,i=0; i<3;)
{
    if(book_buy_id[i]==bk5.book_id)
    {
        dummy[k]=bk5.book_id;
        rectangle(40,d-11,590,d+21);
        dummy[i]=bk5.book_id;
        gotoxy(8,f);
        cout<<bk5.book_name;
        gotoxy(66,f);
        cout<<bk5.price;
        gotoxy(42,f);
        cout<<bk5.author;
        f+=3;
        d+=48;
        filbookread.seekg(0);
        ++i;
        ++k;
    }
    if(book_buy_id[i]==0)
    {
        break;
    }
    filbookread.read((char *) &bk5,sizeof(bk5));
}
filbookread.close();
return_mouse:
mouse();
int re_id=mouse_click_for_display(cor_x,cor_y);
switch(re_id)
{
case 0:
case 1:
case 2:
    return_id=dummy[re_id];
    break;
case 6:
    exit_choice=exit_confirm();
    if(exit_choice=='y')
        return;
    else if(exit_choice=='n')
        goto return_book;
default:
    goto return_mouse;
}
if(return_id==0)
    goto return_book;
for(i=0; i<3; ++i)
{
    if(return_id==book_buy_id[i])

```

```

        {
            book_issue_index=i;
            char sell_conf= sell_confirm();
            if(sell_conf=='n')
                goto return_book;
        }
    }
    ifstream filbookmatch;
    filbookmatch.open("book",ios::in|ios::binary);
    if(!filbookmatch)
    {
        cout<<"cannot open file!!";
        return ;
    }
    filbookmatch.seekg(0);
    filbookmatch.read((char *) &bk5,sizeof(bk5));
    while(filbookmatch)
    {
        if(bk5.book_id==return_id)
        {
            pos=filbookmatch.tellg();
            break;
        }
        filbookmatch.read((char *) &bk5,sizeof(bk5));
    }
    filbookmatch.close();
    ofstream filbooksell;
    filbooksell.open("book",ios::in|ios::binary);
    if(!filbooksell)
    {
        cout<<"cannot open file!!";
        return ;
    }
    book_buy_id[book_issue_index]=0;
    for(i=book_issue_index; i<2; ++i)
    {
        book_buy_id[i]=book_buy_id[i+1];
    }
    book_buy_id[2]=0;
    bk5.buy_stock+=1;
    amount=amount+(bk5.price*0.8);
    filbooksell.seekp(pos+sizeof(bk5));
    filbooksell.write((char *) &bk5,sizeof(bk5));
    filbooksell.close();
    goto start;
}
/*****
*                               END OF CLASS DEFINATION                               *
*****/
user signin(void);
void admin_home_page(admin ad);
void premium_home_page(premium pre);
void regular_home_page(regular reg);
user us;
admin ad;
regular reg;
premium pre;
book bk;
/*****
*                               BODY OF MAIN                               *
*****/
void main()
{
    int member;
    int gdriver=DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    cleardevice();
    setlinestyle(0,1,3);
    setfillstyle(SOLID_FILL,15);
    bar(0,0,640,480);
    setcolor(5);
    setttextstyle(1,0,15);

```

```

    outtextxy(170,100,"HTCL");
    setcolor(9);
    setttextstyle(0,0,2);
    outtextxy(95,300,"Hi-TECH COMPUTERISED LIBRARY");
    mouse();
//    loading();
main_start:
    clrscr();
    cleardevice();
    us=signin();
    cleardevice();
    switch(us.member_type)
    {
    case 0:
        strcpy(ad.user_name,us.user_name);
        strcpy(ad.user_pass,us.user_pass);
        strcpy(ad.contact,us.contact);
        strcpy(ad.dob,us.dob);
        ad.member_type=us.member_type;
        ad.amount=us.amount;
        ad.book_issue_id[0]=us.book_issue_id[0];
        ad.book_issue_id[1]=us.book_issue_id[1];
        ad.book_issue_id[2]=us.book_issue_id[2];
        admin_home_page(ad);
        break;
    case 1:
        strcpy(pre.user_name,us.user_name);
        strcpy(pre.user_pass,us.user_pass);
        strcpy(pre.contact,us.contact);
        strcpy(pre.dob,us.dob);
        pre.member_type=us.member_type;
        pre.amount=us.amount;
        for(int i=0; i<3; ++i)
        {
            pre.book_issue_id[i]=us.book_issue_id[i];
            pre.book_buy_id[i]=us.book_buy_id[i];
        }
        premium_home_page(pre);
        break;
    case 2:
        strcpy(reg.user_name,us.user_name);
        strcpy(reg.user_pass,us.user_pass);
        strcpy(reg.contact,us.contact);
        strcpy(reg.dob,us.dob);
        reg.member_type=us.member_type;
        reg.amount=us.amount;
        reg.book_issue_id[0]=us.book_issue_id[0];
        reg.book_issue_id[1]=us.book_issue_id[1];
        reg.book_issue_id[2]=us.book_issue_id[2];
        regular_home_page(reg);
        break;
    }
    goto main_start;
}
/*****
*                               END OF MAIN                               *
*****/
/*****
*                               HOME PAGE                                *
*****/
void admin_home_page(admin ad)
{
    int mouse_check_ad=-1;
    char member[10];
user_homestart://7
    clrscr();
    cleardevice();
    ubuntu(232,280);
    setcolor(YELLOW);
    setfillstyle(SOLID_FILL,4);
    bar(9,0,79,80);
    bar(85,0,155,80);

```

```

bar(161,0,231,80);
bar(237,0,307,80);
bar(313,0,383,80);
bar(389,0,459,80);
setfillstyle(SOLID_FILL,14);
bar(464,1,639,189);
side_bar();
setcolor(14);           //CREDITS
logo(44,35,1,1);
outtextxy(14,70,"CREDITS");
setfillstyle(SOLID_FILL,14);
bar(107,10,132,50);           //issue
bar(143,40,148,65);
bar(137,50,154,55);
outtextxy(90,70,"ADD BOOK");
bar(183,10,208,50);           //delbook
line(212,55,228,67);
line(212,67,228,55);
outtextxy(165,70,"DEL BOOK");
circle(272,12,10);           //remove
circle(269,11,1);           //eye
circle(275,11,1);
line(272,22,272,52);           //body
line(272,28,262,38);           // lh
line(272,28,282,38);           //rh
line(272,52,262,60);           //ll
line(272,52,282,60);           //rl
setcolor(10);
line(287,55,305,67);
line(287,67,305,55);
setcolor(14);
outtextxy(242,70,"DEL USER");
setcolor(14);
circle(348,30,20);
line(348,40,348,60);
setfillstyle(SOLID_FILL,4);
bar(342,40,347,65);
bar(349,40,353,65);
outtextxy(317,70,"SIGN OUT");
outtextxy(400,70,"SETTING");
setting_button(422,38);
setfillstyle(SOLID_FILL,0);
bar(480,20,620,170);
logo(550,95,5,3);           //logo
line(460,190,639,190);
gotoxy(59,15);
cout<<"USER_NAME : "<<ad.user_name;
gotoxy(59,18);
cout<<"MEMBERSHIP:Admin";
gotoxy(59,21);
cout<<"AMOUNT      : "<<ad.amount;
gotoxy(59,24);
cout<<"DOB          : "<<ad.dob;
mouse_ad:
mouse();
mouse_check_ad=mouse_click_for_admin_home(cor_x,cor_y);
switch(mouse_check_ad)
{
case 1:
    ad.credits();           //CREDITS
    break;
case 2:
    ad.bookadd();           //ADD BOOK
    break;
case 3:
    ad.delete_book();       //DELETE BOOK
    break;
case 4:
    ad.delete_user();       //DELETE USER
    break;
case 5:
    goto save_ad;

```

```

    case 6:
        ad.setting();
        break;
    default:
        goto mouse_ad;
}
settextstyle(0,0,1);
goto user_homestart;
save_ad:
    //FINAL MODIFICATION AT END
    admin add;
    fstream userdata;
    userdata.open("user",ios::in|ios::out|ios::binary);
    if(!userdata)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    userdata.seekg(0);
    while(userdata)
    {
        userdata.read((char *) &add,sizeof(add));
        if(strcmp(add.user_name,add.user_name)==0)
        {
            int pos;
            pos=userdata.tellg();
            userdata.seekp(pos-sizeof(add));
            userdata.write((char *) &add,sizeof(add));
            break;
        }
    }
    userdata.close();
}
void premium_home_page(premium pre)
{
    int mouse_check_pre=-1,i=0,j=0,book_id_display;
    char member[10];
    book bk_pre,bk12;
user_homestart:
    clrscr();
    cleardevice();
    setfillstyle(SOLID_FILL,15);          //background
    bar(3,83,459,480);
    setfillstyle(SOLID_FILL,4);
    bar(5,0,71,80);
    bar(76,0,136,80);
    bar(141,0,201,80);
    bar(206,0,266,80);
    bar(271,0,331,80);
    bar(336,0,399,80);
    bar(404,0,464,80);
    setfillstyle(SOLID_FILL,14);
    bar(464,1,639,189);
    side_bar();
    setcolor(14);
    logo(44,35,1,1);
    outtextxy(14,70,"CREDITS");
    setfillstyle(SOLID_FILL,14);
    bar(85,10,112,50);          //issue
    bar(125,40,122,65);
    bar(115,50,132,55);
    outtextxy(86,70,"ISSUE");
    bar(150,10,175,50);          //return
    bar(180,55,195,60);
    outtextxy(148,70,"RETURN");
    bar(219,25,254,55);          //buy
    rectangle(229,15,244,25);
    outtextxy(225,70,"BUY");
    bar(284,18,319,45);
    rectangle(294,45,309,58);
    outtextxy(286,70,"SELL");

```

```

circle(366,30,20); //sign_out
line(366,40,366,60);
setfillstyle(SOLID_FILL,4);
bar(360,40,365,65);
bar(367,40,372,65);
outtextxy(337,70,"SIGN_OUT");
outtextxy(408,70,"SETTING");
setting_button(435,38);
setfillstyle(SOLID_FILL,0);
setcolor(0);
setlinestyle(SOLID_LINE,1,1);
outtextxy(190,90,"ISSUED BOOKS"); // ISSUED BOOKS
line(0,100,462,100);
setfillstyle(SOLID_FILL,0);
bar(8,110,455,160); //bar1
bar(8,165,455,215); //bar2
bar(8,220,455,270); //bar3
line(0,280,462,280);
outtextxy(190,285,"BUYED BOOKS"); // BUYED BOOKS
line(0,300,462,300);
setlinestyle(0,1,3);
setfillstyle(SOLID_FILL,0);
bar(8,310,455,360); //bar4
bar(8,365,455,415); //bar5
bar(8,420,455,470); //bar6
setcolor(14);
for(book_id_display=0,i=130,j=330; book_id_display<3;
    ++book_id_display,i+=55,j+=55)
{
    ifstream book_read;
    book_read.open("book",ios::in|ios::binary);
    if(!book_read)
    {
        cout<<"cannot open file!!";
        return ;
    }
    book_read.seekg(0);
    book_read.read((char *) &bk12,sizeof(bk12));
    while(book_read)
    {
        if(bk12.book_id==pre.book_issue_id[book_id_display])
        {
            settextstyle(0,0,1);
            outtextxy(25,i,bk12.book_name);
        }
        if(bk12.book_id==pre.book_buy_id[book_id_display])
        {
            settextstyle(0,0,1);
            outtextxy(25,j,bk12.book_name);
            // break;
        }
        book_read.read((char *) &bk12,sizeof(bk12));
    }
    book_read.close();
}
setcolor(15);
gotoxy(59,15);
cout<<"USER_NAME : "<<pre.user_name;
gotoxy(59,18);
cout<<"MEMBERSHIP:Premium";
gotoxy(59,21);
cout<<"AMOUNT : "<<pre.amount;
gotoxy(59,24);
cout<<"DOB : "<<pre.dob;
mouse_pre:
mouse();
mouse_check_pre=mouse_click_for_premium_home(cor_x,cor_y);
switch(mouse_check_pre)
{
case 1:
    bk_pre.read_book(pre.book_issue_id[0]);
    goto read_book;

```

```

    case 2:
        bk_pre.read_book(pre.book_issue_id[1]);
        goto read_book;
    case 3:
        bk_pre.read_book(pre.book_issue_id[2]);
        goto read_book;
    case 4:
        bk_pre.read_book(pre.book_buy_id[0]);
        goto read_book;
    case 5:
        bk_pre.read_book(pre.book_buy_id[1]);
        goto read_book;
    case 6:
        bk_pre.read_book(pre.book_buy_id[2]);
        goto read_book;
    case 7:
        pre.credits(); //CREDITS
        break;
    case 8:
        pre.issue_book(); //ISSUE BOOK
        break;
    case 9:
        pre.return_book(); //RETURN BOOK
        break;
    case 10:
        pre.buy_book(); //BUY BOOK
        break;
    case 11:
        pre.sell_book(); //sell book
        break;
    case 12:
        goto save_pre;
    case 13:
        pre.setting();
        break;
    default:
        goto mouse_pre;
}
read_book:
    goto user_homestart;
save_pre:
    //FINAL MODIFICATION AT END
    premium pr;
    fstream userdata;
    userdata.open("user",ios::in|ios::out|ios::binary);
    if(!userdata)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    userdata.seekg(0);
    while(userdata)
    {
        userdata.read((char *) &pr,sizeof(pr));
        if(strcmp(pre.user_name,pr.user_name)==0)
        {
            int pos;
            pos=userdata.tellg();
            userdata.seekp(pos-sizeof(pre));
            userdata.write((char *) &pre,sizeof(pre));
            break;
        }
    }
    userdata.close();
}
void regular_home_page(regular reg)
{
    char member[10];
    int check_mouse2=-1,i=0,book_id_display;
    book bk_reg,bk12;
    user_homestart:

```



```

clrscr();
cleardevice();
setcolor(YELLOW);
setfillstyle(SOLID_FILL,15);
bar(3,83,459,480); //background
setfillstyle(SOLID_FILL,4);
bar(5,0,85,80);
bar(95,0,175,80);
bar(185,0,275,80);
bar(285,0,365,80);
bar(375,0,455,80);
setcolor(14); //CREDITS
logo(44,35,1,1);
outtextxy(14,70,"CREDITS");
side_bar();
setfillstyle(SOLID_FILL,14);
bar(120,10,145,50); //issue
bar(160,40,157,65);
bar(150,50,167,55);
outtextxy(95,70,"ISSUE BOOK");
bar(217,10,242,50); //return
bar(247,55,262,60);
outtextxy(188,70,"RETURN BOOK");
setcolor(14);
circle(325,30,20); //sign_out
line(325,40,325,60);
setfillstyle(SOLID_FILL,4);
bar(320,40,324,65);
bar(326,40,330,65);
outtextxy(295,70,"SIGN OUT");
outtextxy(388,70,"SETTING");
setting_button(415,38);
setfillstyle(SOLID_FILL,0);
logo(550,95,5,3); //logo
setcolor(0);
line(460,190,639,190); //issue
setlinestyle(SOLID_LINE,1,1);
outtextxy(190,90,"ISSUED BOOKS");
line(0,100,462,100);
bar(8,110,453,220); //bar1
bar(8,230,453,340); //bar2
bar(8,350,453,460); //bar3
setlinestyle(0,1,3);
setcolor(14);
for(book_id_display=0,i=160; book_id_display<3; ++book_id_display,i+=120)
{
    ifstream book_read;
    book_read.open("book",ios::in|ios::binary);
    if(!book_read)
    {
        cout<<"cannot open file!!";
        return ;
    }
    book_read.seekg(0);
    book_read.read((char *) &bk12,sizeof(bk12));
    while(book_read)
    {
        if(bk12.book_id==reg.book_issue_id[book_id_display])
        {
            settextstyle(0,0,1);
            outtextxy(25,i,bk12.book_name);
            book_read.seekg(0);
            break;
        }
        book_read.read((char *) &bk12,sizeof(bk12));
    }
    book_read.close();
}
setcolor(15);
gotoxy(59,15);
cout<<"USER_NAME : "<<reg.user_name;
gotoxy(59,18);

```

```

cout<<"MEMBERSHIP:Regular";
gotoxy(59,21);
cout<<"AMOUNT      : "<<reg.amount;
gotoxy(59,24);
cout<<"DOB          : "<<reg.dob;
//working starts::::MENU
mouse_homeuser:
mouse();
check_mouse2=mouse_click_for_regular_home(cor_x,cor_y);
switch(check_mouse2)
{
case 1:
    bk_reg.read_book(reg.book_issue_id[0]);
    goto read_book;
case 2:
    bk_reg.read_book(reg.book_issue_id[1]);
    goto read_book;
case 3:
    bk_reg.read_book(reg.book_issue_id[2]);
    goto read_book;
case 4:
    reg.credits();           //CREDITS
    break;
case 5:
    reg.issue_book();       //ISSUE BOOK
    break;
case 6:
    reg.return_book();      //RETURN BOOK
    break;
case 7:
    goto end_home;         //SIGN OUT
case 8:
    reg.setting();         //SETTINGS
    break;//settings
default:
    goto mouse_homeuser;
}
read_book:
    goto user_homestart;
end_home:
    //FINAL MODIFICATION AT END
    regular urs;
    fstream userdata;
    userdata.open("user",ios::in|ios::out|ios::binary);
    if(!userdata)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return ;
    }
    userdata.seekg(0);
    while(userdata)
    {
        userdata.read((char *) &urs,sizeof(urs));
        if(strcmp(reg.user_name,urs.user_name)==0)
        {
            int pos;
            pos=userdata.tellg();
            userdata.seekp(pos-sizeof(reg));
            userdata.write((char *) &reg,sizeof(reg));
            break;
        }
    }
    userdata.close();
}
/*****
*                               END OF HOMEPAGE                               *
*****/
/*****
*                               FUNCTION DEFINATION                             *
*****/
void side_bar()

```

```

{
    setfillstyle(SOLID_FILL,8);
    bar(464,1,639,189);
    setfillstyle(SOLID_FILL,0);
    bar(480,20,620,170);
    logo(550,95,5,3); //logo
    setfillstyle(SOLID_FILL,8); // gap bar
    bar(464,195,639,223);
    bar(464,240,639,268);
    bar(464,288,639,318);
    bar(464,337,639,367);
    bar(464,385,639,415);
    bar(464,440,639,470);
    setcolor(15);
    settextstyle(1,0,2);
    outtextxy(530,415,"HTCL");
    settextstyle(0,0,1);
    setcolor(0);
}
void ubuntu(int xu,int yu)
{
    //////////////////////////////////////////circle1
    setcolor(14);
    setfillstyle(1,14);
    sector(xu-10,yu,230,110,96,100);
    //////////////////////////////////////////circle2
    setcolor(4);
    setfillstyle(1,4);
    sector(xu,yu+10,230,350,100,100);
    //////////////////////////////////////////circle3
    setcolor(12);
    setfillstyle(1,12);
    sector(xu+3,yu-5,0,110,100,100);
    sector(xu+3,yu-5,351,360,100,100);
    setcolor(0);
    //////////////////////////////////////////black
    setfillstyle(1,0);
    sector(xu,yu,0,360,80,80);
    //////////////////////////////////////////smallcircle1
    sector(xu-120,yu-10,0,360,25,25);
    setcolor(4);
    setfillstyle(1,4);
    sector(xu-120,yu-10,0,360,15,15);
    //////////////////////////////////////////smallcircle2
    setcolor(0);
    setfillstyle(1,0);
    sector(xu+35,yu+117,0,360,25,25);
    setcolor(12);
    setfillstyle(1,12);
    sector(xu+35,yu+117,0,360,15,15);
    //////////////////////////////////////////smallcircle3
    setcolor(0);
    setfillstyle(1,0);
    sector(xu+65,yu-100,0,360,25,25);
    setcolor(14);
    setfillstyle(1,14);
    sector(xu+65,yu-100,0,360,15,15);
}
void wrong_password(void)
{
    clrscr();
    cleardevice();
    setfillstyle(SOLID_FILL,4);
    bar(180,140,420,290);
    settextstyle(0,0,9);
    setcolor(15);
    outtextxy(185,180,"!");
    settextstyle(0,0,2);
    outtextxy(285,195,"WRONG");
    outtextxy(260,220,"PASSWORD");
    settextstyle(0,0,1);
    mouse();
}

```

```

}
user_signin(void)
{
    user ur;
    int check_pass,check_save;
    int check=-1,check1=-1,check_mouse=-1;
    char dob[15],username[20],password[20],exitc=NULL;
refill:
    check_mouse=-1;
    check1=-1;
    clrscr();
    cleardevice();
    setcolor(4);
    setfillstyle(SOLID_FILL,7);           //SIGNIN BOX
    bar(50,95,130,380);
    settextstyle(1,0,8);
    outtextxy(65,80,"H");
    outtextxy(67,150,"T");
    outtextxy(69,220,"C");
    outtextxy(72,290,"L");
    settextstyle(0,0,0);
    logo(290,100,6,4);
    setfillstyle(SOLID_FILL,14);         //SIGNIN BOX
    bar(160,200,400,350);
    setfillstyle(SOLID_FILL,0);          //USERNAME
    bar(255,222,390,240);               //PASSWORD
    bar(255,255,390,273);
    setfillstyle(SOLID_FILL,7);          //SIDEBAR
    bar(499,0,639,479);
    setfillstyle(SOLID_FILL,4);          //SIGN UP BUTTON
    bar(250,300,310,325);               //CREATE ACCOUNT
    bar(499,215,639,235);               //FORGOT PASSWORD
    bar(499,265,639,285);               //EXIT
    bar(499,315,639,335);
    setcolor(14);
    outtextxy(252,310,"SIGN_IN");
    setcolor(0);
    outtextxy(168,228,"User_name");
    outtextxy(168,261,"Password");
    outtextxy(513,222,"CREATE ACCOUNT");
    outtextxy(512,272,"FORGOT PASSWORD");
    outtextxy(550,322,"EXIT");
mouse_signin:
    mouse();
    check_mouse=mouse_click_for_signin(cor_x,cor_y);
    switch(check_mouse)
    {
    case 1:
        gotoxy(33,15);
        cin.getline(username,20);
        gotoxy(33,17);
        cin.getline(password,20);
        goto mouse_signin;
    case 2:
        goto out;
    case 3:
        check_save=ur.get_data();
        if(check_save==0)
        {
            ofstream filout;
            filout.open("user",ios::app|ios::binary);
            filout.write((char *) &ur,sizeof(ur));
            filout.close();
        }
        goto refill;
    case 4:
    {
        setfillstyle(SOLID_FILL,0);       //SIDEBAR
        bar(499,0,639,479);
        setfillstyle(SOLID_FILL,4);       //ENTER BUTTON
        bar(250,300,310,325);
        setcolor(15);
    }
    }
}

```

```

        outtextxy(260,310,"ENTER");
        setcolor(0);
        outtextxy(168,228,"User_name");
        setfillstyle(SOLID_FILL,14);
        bar(160,260,250,280);
        outtextxy(170,261,"DOB");
se_signin:
    mouse();
    int check=mouse_click_for_signin(cor_x,cor_y);
    if(check!=1)
        goto se_signin;
    gotoxy(33,15);
    cin.getline(username,20);
    gotoxy(33,17);
    cin.getline(dob,15);
again:
    mouse();
    int check5=mouse_click_for_signin(cor_x,cor_y);
    if(check5!=2)
        goto again;
    ifstream filinf;
    filinf.open("user",ios::in|ios::binary);
    if(!filinf)
    {
        cout<<"cannot open file!!";
        exit(0);
    }
    filinf.seekg(0);
    while(filinf)
    {
        filinf.read((char *) &ur,sizeof(ur));
        if(strcmp(username,ur.user_name)==0&&strcmp(dob,ur.dob)==0)
        {
            check1=0;
            break;
        }
    }
    filinf.close();
    if(check1==0)
    {
        cleardevice();
        setfillstyle(SOLID_FILL,7);
        bar(180,120,420,300);
        setcolor(0);
        setttextstyle(0,0,2);
        outtextxy(250,150,"YOUR");
        outtextxy(210,200,"PASSWORD IS");
        setcolor(9);
        outtextxy(240,260,ur.user_pass);
        delay(3000);
        setttextstyle(0,0,1);
        cleardevice();
        setcolor(15);
        goto refill;
    }
}
break;
case 5 :
    exitc=exit_confirm();
    if(exitc=='y' || exitc=='Y')
        exit(0);
    else
        goto refill;
default:
    goto mouse_signin;
}
out:
    cleardevice();
    ifstream filin;
    filin.open("user",ios::in|ios::binary);
    if(!filin)
    {

```

```

        cout<<"cannot open file!!";
        delay(2000);
        exit(0);
    }
    filin.seekg(0);
    while(filin)
    {
        filin.read((char *) &ur,sizeof(ur));
        if(strcmp(username,ur.user_name)==0&&strcmp(password,ur.user_pass)==0)
        {
            signin_sucess();
            check=0;
            break;
        }
    }
    if(check!=0)
    {
        wrong_password();
        goto refill;
    }
    filin.close();
    return ur;
}
int book_display(int operation)
{
    book bk8;
    int last=0,i,buk_issue_id,pos=1,buk_id;
    char exit_choice=NULL;
re_display:
    clrscr();
    cleardevice();
    last=0;
    ifstream filebookdisp;
    filebookdisp.open("book",ios::in|ios::binary);
    if(! filebookdisp)
    {
        cout<<"cannot open file!!";
        delay(2000);
        return 0;
    }
    outtextxy(70,100,"BOOK NAME");
    outtextxy(350,100,"AUTHOR");
    outtextxy(510,100,"PRICE");
    setfillstyle(SOLID_FILL, BLUE);
    bar(0,0,639,90);
    settextstyle(0,0,4);
    switch(operation)
    {
    case 1:
        outtextxy(160,30,"ISSUE BOOK");
        break;
    case 2:
        outtextxy(180,30,"BUY BOOK");
        break;
    case 3:
        outtextxy(180,30,"READ BOOK");
        break;
    case 4:
        outtextxy(170,30,"DELETE BOOK");
        break;
    }
    settextstyle(0,0,1);
    exit_button();
    if(pos>4)
    {
        setfillstyle(SOLID_FILL,9);
        bar(39,330,130,360);
        outtextxy(55,342,"PREVIOUS");
    }
    setfillstyle(SOLID_FILL,1);
    bar(510,330,590,360);
    outtextxy(535,342,"NEXT");
}

```

```

filebookdisp.seekg(0);
for(int k=0; k<pos; ++k)
{
    filebookdisp.read((char *) &bk8,sizeof(bk8));
}
setcolor(15);
int dmy[4]= {0,0,0,0};
int d,f=10;
for(d=148,i=0; i<4&&filebookdisp!=0; d+=48,f+=3,++i)
{
    rectangle(40,d-11,590,d+21);
    dmy[i]=bk8.book_id;
    gotoxy(8,f);
    cout<<bk8.book_name;
    gotoxy(65,f);
    cout<<bk8.price;
    gotoxy(42,f);
    cout<<bk8.author;
    filebookdisp.read((char *) &bk8,sizeof(bk8));
    ++pos;
    last++;
}
re_ans:
mouse();
buk_issue_id=mouse_click_for_diply(cor_x,cor_y);
switch(buk_issue_id)
{
case 0:
case 1:
case 2:
case 3:
    if(dmy[buk_issue_id]==0)
        goto re_ans;
    buk_id=dmy[buk_issue_id];
    filebookdisp.close();
    break;
case 4:
    filebookdisp.close();
    if(pos>4)
        pos--=(4+last);
    else
        pos=1;
    goto re_diply;
case 5:
    if( filebookdisp==0)
    {
        pos=1;
        filebookdisp.close();
    }
    goto re_diply;
case 6:
    exit_choice=exit_confirm();
    filebookdisp.close();
    if(exit_choice=='y')
        return -1;
    else if(exit_choice=='n')
    {
        pos-=last;
        goto re_diply;
    }
    break;
default:
    goto re_ans;
}
return buk_id;
}
void setting_button(int xu,int yu)
{
    setcolor(14);
    setfillstyle(1,14);
    sector(xu,yu,0,360,15,19);
    //////////////////////////////////white circle

```

```

        setcolor(4);
        setfillstyle(1,4);
        sector(xu,yu,0,360,7,11);
////////////////////////////////////3
        setcolor(14);
        setfillstyle(1,14);
        sector(xu,yu-21,0,360,2,2);
////////////////////////////////////black
        setcolor(14);
        setfillstyle(1,14);
        sector(xu,yu+21,0,360,2,2);
////////////////////////////////////s1
        setcolor(14);
        setfillstyle(1,14);
        sector(xu-17,yu,0,360,2,2);
////////////////////////////////////s2
        setcolor(14);
        setfillstyle(1,14);
        sector(xu+17,yu,0,360,2,2);
        setcolor(14);
        setfillstyle(1,14);
        sector(xu-12,yu-14,0,360,2,2);
////////////////////////////////////s3
        setcolor(14);
        setfillstyle(1,14);
        sector(xu+12,yu-14,0,360,2,2);
        setcolor(14);
        setfillstyle(1,14);
        sector(xu-12,yu+14,0,360,2,2);
        setcolor(14);
        setfillstyle(1,14);
        sector(xu+12,yu+14,0,360,2,2);
    }
void signin_sucess()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(180,120,420,300);
    setcolor(1);
    settextstyle(0,0,2);
    outtextxy(235,150,"YOU HAVE");
    outtextxy(205,200,"SUCCESSFULLY");
    outtextxy(230,260,"LOGGED IN");
    mouse();
    settextstyle(0,0,1);
    setcolor(15);
}
void issue_sucess()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(180,120,420,290);
    setcolor(1);
    settextstyle(0,0,2);
    outtextxy(235,150,"YOU HAVE");
    outtextxy(205,200,"SUCCESSFULLY");
    outtextxy(180,250,"ISSUED THE BOOK");
    mouse();
    settextstyle(0,0,1);
}
void buy_sucess()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(180,120,420,290);
    setcolor(1);
    settextstyle(0,0,2);
    outtextxy(235,150,"YOU HAVE");
    outtextxy(205,200,"SUCCESSFULLY");
    outtextxy(190,250,"BUYED THE BOOK");
    mouse();
    settextstyle(0,0,1);
}

```



```

}
void return_sucess()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(170,120,450,290);
    setcolor(1);
    settextstyle(0,0,2);
    outtextxy(240,150,"YOU HAVE");
    outtextxy(210,200,"SUCCESSFULLY");
    outtextxy(175,250,"RETURNED THE BOOK");
    mouse();
    settextstyle(0,0,1);
    setcolor(15);
}
char return_confirm()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(200,130,410,280);
    setfillstyle(SOLID_FILL,1);
    bar(230,230,280,260);
    bar(330,230,380,260);
    setcolor(4);
    settextstyle(0,0,1);
    outtextxy(255,150,"Are You Sure ");
    outtextxy(225,190,"to RETURN the book ?");
    outtextxy(245,240,"YES");
    outtextxy(350,240,"NO");
mouse_return:
    mouse();
    char return_choice=mouse_click_for_dialoge_box(cor_x,cor_y);
    if(return_choice!='y' && return_choice!='n')
        goto mouse_return;
    else
        settextstyle(0,0,1);
    return return_choice;
}
void sell_sucess()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(180,120,420,290);
    setcolor(1);
    settextstyle(0,0,2);
    outtextxy(235,150,"YOU HAVE");
    outtextxy(205,200,"SUCCESSFULLY");
    outtextxy(200,250,"SOLD THE BOOK");
    mouse();
    settextstyle(0,0,1);
}
char sell_confirm()
{
    cleardevice();
    setfillstyle(SOLID_FILL,7);
    bar(200,130,410,280);
    setfillstyle(SOLID_FILL,4);
    bar(230,230,280,260);
    bar(330,230,380,260);
    setcolor(0);
    settextstyle(0,0,1);
    outtextxy(255,150,"Are You Sure ");
    outtextxy(225,190,"to SELL the book ?");
    outtextxy(245,240,"YES");
    outtextxy(350,240,"NO");
mouse_sell:
    mouse();
    char sell_choice=mouse_click_for_dialoge_box(cor_x,cor_y);
    if(sell_choice!='y' && sell_choice!='n')
        goto mouse_sell;
    else
        settextstyle(0,0,1);
}

```

```

        return sell_choice;
    }
    char deletebook_confirm()
    {
        cleardevice();
        setfillstyle(SOLID_FILL,7);
        bar(200,130,410,280);
        setfillstyle(SOLID_FILL,4);
        bar(230,230,280,260);
        bar(330,230,380,260);
        setcolor(0);
        settextstyle(0,0,1);
        outtextxy(255,150,"Are You Sure ");
        outtextxy(225,190,"to DELETE the Book ?");
        outtextxy(245,240,"YES");
        outtextxy(350,240,"NO");
    mouse_del:
        mouse();
        char del_choice=mouse_click_for_dialoge_box(cor_x,cor_y);
        if(del_choice!='y'&&del_choice!='n')
            goto mouse_del;
        else
            return del_choice;
    }
    char deleteuser_confirm()
    {
        cleardevice();
        setfillstyle(SOLID_FILL,7);
        bar(200,130,410,280);
        setfillstyle(SOLID_FILL,4);
        bar(230,230,280,260);
        bar(330,230,380,260);
        setcolor(0);
        settextstyle(0,0,1);
        outtextxy(255,150,"Are You Sure ");
        outtextxy(225,190,"to REMOVE the user ?");
        outtextxy(245,240,"YES");
        outtextxy(350,240,"NO");
    mouse_deluser:
        mouse();
        char deluser_choice=mouse_click_for_dialoge_box(cor_x,cor_y);
        if(deluser_choice!='y'&&deluser_choice!='n')
            goto mouse_deluser;
        else
            settextstyle(0,0,1);
            setcolor(15);
            return deluser_choice;
    }
    char exit_confirm()
    {
        cleardevice();
        setfillstyle(SOLID_FILL,7);
        bar(200,130,410,280);
        setfillstyle(SOLID_FILL,4);
        bar(230,230,280,260);
        bar(330,230,380,260);
        setcolor(0);
        settextstyle(0,0,1);
        outtextxy(255,150,"Are You Sure ");
        outtextxy(225,190,"you want to EXIT ?");
        outtextxy(245,240,"YES");
        outtextxy(350,240,"NO");
    mouse_exit:
        mouse();
        char exit_choice=mouse_click_for_dialoge_box(cor_x,cor_y);
        if(exit_choice!='y'&&exit_choice!='n')
            goto mouse_exit;
        else
            return exit_choice;
    }
    void logo(int x,int y,int xradl,int diff)
    {

```

```

setlinestyle(SOLID_LINE,1,0);
int xrad2,xrad3,xrad4,xrad5;
xrad2=xrad1+(3*diff);
xrad3=xrad2+(4*diff);
xrad4=xrad3+(5*diff);
xrad5=xrad4+(6*diff);
setcolor(0);
//FIRST
setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad5, xrad5);
setfillstyle(SOLID_FILL, 6);
sector(x,y, 0, 90, xrad4, xrad4);
setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad3, xrad3);
setfillstyle(SOLID_FILL, 6);
sector(x,y, 0, 90, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad1, xrad1);
//SECOND
setfillstyle(SOLID_FILL, 15);
sector(x, y, 90, 180, xrad5, xrad5);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad4, xrad4);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 90, 180, xrad3, xrad3);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad1, xrad1);
//THIRD
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad5, xrad5);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 180, 270, xrad4, xrad4);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad3, xrad3);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 180, 270, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad1, xrad1);
//FOURTH
setfillstyle(SOLID_FILL, 15);
sector(x, y, 270, 360, xrad5, xrad5);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad4, xrad4);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 270, 360, xrad3, xrad3);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad1, xrad1);
setlinestyle(0,1,3);
}
void loading(void)
{
    clrscr();
    cleardevice();
    setlinestyle(SOLID_LINE,1,0);
    int x=320,y=200,xrad1=10,diff=5,xrad2,xrad3,xrad4,xrad5;
    xrad2=xrad1+(3*diff);
    xrad3=xrad2+(4*diff);
    xrad4=xrad3+(5*diff);
    xrad5=xrad4+(6*diff);
    delay(500);
    //FIRST
    for(int i=0; i<3; ++i)
    {
        cleardevice();
        setcolor(2);
        setttextstyle(0,0,2);
        outtextxy(235,380,"LOADING");
        setcolor(0);
    }
}

```

```

setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad5, xrad5);
setfillstyle(SOLID_FILL, 6);
sector(x,y, 0, 90, xrad4, xrad4);
setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad3, xrad3);
setfillstyle(SOLID_FILL, 6);
sector(x,y, 0, 90, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x,y, 0, 90, xrad1, xrad1);
delay(500);
//SECOND
setcolor(4);
settextstyle(0,0,2);
outtextxy(345,380,".");
setcolor(0);
setfillstyle(SOLID_FILL, 15);
sector(x, y, 90, 180, xrad5, xrad5);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad4, xrad4);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 90, 180, xrad3, xrad3);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 90, 180, xrad1, xrad1);
delay(500);
//THIRD
setcolor(4);
settextstyle(0,0,2);
outtextxy(355,380,".");
setcolor(0);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad5, xrad5);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 180, 270, xrad4, xrad4);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad3, xrad3);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 180, 270, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 180, 270, xrad1, xrad1);
delay(500);
//FOURTH
setcolor(4);
settextstyle(0,0,2);
outtextxy(365,380,".");
setcolor(0);
setfillstyle(SOLID_FILL, 15);
sector(x, y, 270, 360, xrad5, xrad5);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad4, xrad4);
setfillstyle(SOLID_FILL, 6);
sector(x, y, 270, 360, xrad3, xrad3);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad2, xrad2);
setfillstyle(SOLID_FILL, 0);
sector(x, y, 270, 360, xrad1, xrad1);
delay(500);
}
setlinestyle(0,1,3);
}
void exit_button()
{
    setfillstyle(SOLID_FILL,4);
    bar(600,0,639,20);
    setcolor(15);
    settextstyle(0,0,1);
    outtextxy(615,8,"X");
}
/*****
*
* FUNCTION TO CHECK MOUSE COORDINATES
*****/

```

```

*****/
int mouse_click_for_setting(int a,int b)
{
    if(a>=400&&a<639&&b>35&&b<65)
        return 1;
    else if(a>=400&&a<639&&b>85&&b<115)
        return 2;
    else if(a>=400&&a<639&&b>135&&b<165)
        return 3;
    else if(a>=400&&a<639&&b>185&&b<215)
        return 4;
    else if(a>=400&&a<639&&b>235&&b<265)
        return 5;
    else if(a>=400&&a<639&&b>285&&b<315)
        return 6;
    else if(a>=600&&a<639&&b>0&&b<20)
        return 7;
    else
        return 0;
}

int mouse_click_for_signin (int a,int b)
{
    if(a>=255 && a<390 && b>220&& b<241)
        return 1;
    else if(a>=250 && a<310 && b>300&& b<325)
        return 2;
    else if(a>=499 && a<639 && b>215&& b<235)
        return 3;
    else if(a>=499 && a<640 && b>265&& b<285)
        return 4;
    else if(a>=499 && a<639 && b>315&& b<335)
        return 5;
    else
        return 0;
}

char mouse_click_for_dialoge_box (int a,int b)
{
    if(a>=230 && a<280 && b>230&& b<260)
        return 'y';
    else if(a>=330 && a<380 && b>230&& b<260)
        return 'n';
    else
        return 'r';
}

int mouse_click_for_signup (int a,int b)
{
    if(a>=275 && a<520 && b>94&& b<112)
        return 1;
    else if(a>=600 && a<639 && b>0&& b<20)
        return 2;
    else if(a>=240 && a<400 && b>400&& b<430)
        return 3;
    else
        return 0;
}

int mouse_click_for_diplay(int a,int b)
{
    if(a>=40 && a<590 && b>=137&& b<=169)
        return 0;
    else if(a>=40 && a<590 && b>=185&& b<=217)
        return 1;
    else if(a>=40 && a<590 && b>=233&& b<=265)
        return 2;
    else if(a>=40 && a<590 && b>=281&& b<=313)
        return 3;
    else if(a>=39 && a<130 && b>=330&& b<=360)
        return 4;
    else if(a>=510 && a<590 && b>=330&& b<=360)
        return 5;
    else if(a>=600 && a<639 && b>0&& b<20)
        return 6;
    else

```

```

        return -1;
    }
int mouse_click_for_readbook(int a,int b)
{
    if(a>=600 && a<639 && b>0&& b<20)
        return 1;
    else
        return -1;
}
int mouse_click_for_add_book(int a,int b)
{
    if(a>=80 && a<505 && b>94&& b<112)
        return 1;
    else if(a>=600 && a<639 && b>0&& b<20)
        return 2;
    else if(a>=240 && a<440 && b>400&& b<470)
        return 3;
    else
        return 0;
}
int mouse_click_for_regular_home(int a,int b)
{
    bar(12,110,460,220);
    bar(12,230,460,340);
    bar(12,350,460,460);
    if(a>=12 && a<460 && b>110&& b<220)
        return 1;
    else if(a>=12 && a<460 && b>230&& b<340)
        return 2;
    else if(a>=12 && a<460 && b>350&& b<460)
        return 3;
    else if(a>=5 && a<85 && b>0&& b<80)
        return 4;
    else if(a>=95 && a<175 && b>0&& b<80)
        return 5;
    else if(a>=185 && a<275 && b>0&& b<80)
        return 6;
    else if(a>=285 && a<365 && b>0&& b<80)
        return 7;
    else if(a>=375 && a<455 && b>0&& b<80)
        return 8;
    else
        return 0;
}
int mouse_click_for_premium_home(int a,int b)
{
    if(a>=12 && a<460 && b>110&& b<160)
        return 1;
    else if(a>=12 && a<460 && b>165&& b<215)
        return 2;
    else if(a>=12 && a<460 && b>220&& b<270)
        return 3;
    else if(a>=12 && a<460 && b>310&& b<360)
        return 4;
    else if(a>=12 && a<460 && b>365&& b<415)
        return 5;
    else if(a>=12 && a<460 && b>420&& b<470)
        return 6;
    else if(a>=5 && a<71 && b>0&& b<80)
        return 7;
    else if(a>=76 && a<136 && b>0&& b<80)
        return 8;
    else if(a>=141 && a<201 && b>0&& b<80)
        return 9;
    else if(a>=206 && a<271 && b>0&& b<80)
        return 10;
    else if(a>=271 && a<336 && b>0&& b<80)
        return 11;
    else if(a>=336 && a<399 && b>0&& b<80)
        return 12;
    else if(a>=404 && a<455 && b>0&& b<80)
        return 13;
}

```

```

        else
            return 0;
    }
int mouse_click_for_admin_home(int a,int b)
{
    if(a>=9 && a<79 && b>0&& b<80)
        return 1;
    else if(a>=85 && a<155 && b>0&& b<80)
        return 2;
    else if(a>=161 && a<231 && b>0&& b<80)
        return 3;
    else if(a>=237 && a<307 && b>0&& b<80)
        return 4;
    else if(a>=313 && a<383 && b>0&& b<80)
        return 5;
    else if(a>=389 && a<459 && b>0&& b<80)
        return 6;
    else
        return 0;
}
/*****
*           END OF FUNCTION TO CHECK MOUSE CORDINATES           *
*****/
/*****
*           END OF FUNCTION DEFINATION                           *
*****/
/*****
*           END OF PROGRAM                                       *
*****/

```