COMPUTER SCIENCE PROJECT

E-BOOK MANAGEMENT SYSTEM

R&D BY: YASH TYAGI

Acknowledgement

We would like to express my special thanks of gratitude to my teacher **Mr. Farroukh Nadeem** who gave us the golden opportunity to do this wonderful project on the topic **E-BOOK MANAGEMENT SYSTEM** which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them.

Secondly we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

Thank You.

Certificate

This is to certify that **Yash Tyagi** has successfully completed the project on the topic **E-Book Management System** under the guidance of **Mr. Farroukh Nadeem.**

This project is absolutely genuine does not indulge in plagiarism of any kind. The reference taken in making this project has been declared at the end of this project.

CONTENTS

SYSTEM DESCRIPTION AT A GLANCE

The E-Book Management System automates the library functions to Online in the day-to-day operations of a library. It supports functions such as is download the functions of searching for a particular book, etc.

It also maintains data about books about books and user records that are required during various E-Book operations. The software aims to make the system user friendly and efficient.

The functions that the E-Book Management System provides are as follows:

1. INSERT: This operation is performed when new data needs to be added to the system. This allows entering data about newly purchased eBook's into the books database. The data entered includes book's author, title, publisher and various other fields provided in the form. The data must be accurate and must be entered in the correct format as indicated in the forms. The book will only be add by the administrative only.

2. DELETE: This operation clears the existing records in the various databases. It is used when for e.g. a member leaves college or when book is disposed of from library

a) Book: This will enter a null value for the book whose accession number is entered in the field provided in the respected form. The will only be deleted my admin only not my any member.

b) User: This will clear the record for the particular user whose record needs to be deleted by entering its member no. in the required field. The member can be deleted my member itself not by anyone else.

3. UPDATE: This function updates data in the various records.

a) Book: This function generally would not be required for updating a book as that data wouldn't change.

b) User: This will update the data of particular user, whose data has changed like name address, phone number, etc. by entering her member number.

4. SEARCH: This function is used to search particular data from the database. To search for a particular book. This can be done by entering value in any one or more fields in the form to perform the search such as title or author name.

5. Download: This operation is used for downloading a book to a member of the library. The system automatically downloads the book .The software only share's the link of the book

not whole book. The link in the HTML code so user cannot find the book .The book will be read only so user will not be able to change it or share it with other.

6. DISPLAY: This is used to display each and every record, i.e. record of every book and user in the library software.

a) Book: Record of every book, i.e. its accession number, author name, publisher name, etc.

b) User: Record of every user, i.e. its member code, name, address, phone no.

7. EXIT: This takes user out of the application.

DATA DESIGN

The impact of data structure on program structure and procedural complexities lead to data design to have a profound influence on system quality. All the data objects required by this system are listed below:

1. member.dat :

   The member.dat is to be created on the basis of information collected from the member. It contains information related to:

   - Member code
   - Member name
   - Member address
   - Member phone number

2. book.dat :

   The book.dat is to be created on the basis of information collected from the member. It contains information related to:

   - Book code
   - Book name
   - Book auther name
   - Book publisher
   - Book location

3. temp.dat :

   The temp.dat is to be created on the basis of information collected from the member. It used for:

   - Deleting in record
   - In identification
   - To modify record

4. download.txt:

   The download.txt is to be created on the basis of information collected from the user system. It used for storing deletion date of the book.

FUNCTION DESCRIPTION

- MENU Class Modulation:

S.No.	Function Name	Description
1.	void main_menue(void)	to display main menu
2.	void admin(void)	to display administrative menu
3.	void membermenue(void)	to display member menu
4.	void intro(void)	to display about the project
5.	void edit_book(void)	to display edit menu of book
6.	void edit_member(void)	to display edit menu of member

- BOOK Class Modulation:

S.No.	Function Name	Description
1.	char *bookaddr(int)	Function to return book address
2.	void list(void)	Function to display the list of books
3.	int book_found(int)	Function to return book code
4.	int bookname_found(…)	Function to return book name
5.	int recordno(int)	Function to return last record code
6.	int bookcodeof(char[])	Function to return book code
7.	int reccount(void)	Function to check for empty file
8.	char *bookname(int)	Function to return book name
9.	char *authorname(int)	Function to return author name
10.	char *bookpublisher(int)	Function to return book publisher
11.	void add_new_book(…])	Function to add new record to file
12.	void identify_book(int)	Function to find the given code book
13.	void modify(…)	Function to modify the book record
14.	void deletion(void)	Function to delete a record
15.	void display(int)	Function to display given code book
16.	void delete_rec(int)	Function to delete a record

- MEMBER Class Modulation:

S.No.	Function Name	Description
1.	char *membername(int)	Function to return member name
2.	void list(void)	Function to display all members
3.	int member_found(int)	Function to return member of given code
4.	int lastcode(void)	Function to return last code of member
5.	char *memberphone(int)	Function to return member phone no.

6.	char *memberaddress(int)	Function to return member address
7.	void add_mem(…)	Function to add a new record
8.	void modify(…)	Function to modify a record
9.	void update_book(int, int)	Function to update the record
10.	void display(int)	Function to display the record
11.	void delete_rec(int)	Function to delete the record

➢ WORKING Class Modulation:

S.No.	Function Name	Description
1.	void downloadbook(void)	Function to download the bood
2.	void add_book(void)	Function to add a book
3.	void add_member(void)	Function to add the member
4.	void modify_book(void)	Function to modify the book
5.	void modify_member(void)	Function to modify the member
6.	void delete_book(void)	Function to delete the book
7.	void delete_member(void)	Function to delete the member
8.	void searchbook(void)	Function to search a book

➢ Other Programs

S.No.	Function Name	Description
1.	void password()	Function to enter password for admin Menu

➢ Header file function

S.No.	Function Name	Description
1.	void link(…)	Function to create link of book
2.	void storedate(void)	Function to get date of user system
3.	void deldate(…)	Function to store the delete date of book
4.	void checkdate(void)	Function to check for the deletion of book
5.	void colorbox(…)	Function to make color box

Source Code

```
//********************************************************
*********************

// PROJECT ONLINE BOOK LIBRARY

//********************************************************
*********************



//********************************************************
*********************

// INCLUDED HEADER FILES

//********************************************************
*********************

#include <fstream.h>

#include <string.h>

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include <ctype.h>

#include <dos.h>

#include <colbox.h> //USER DEFINED HEADER FILE

#include <date.h>  //USER DEFINED HEADER FILE
```

```
//********************************************************
*********************

// THIS CLASS CONTROL ALL THE FUNCTIONS IN THE MENU

//********************************************************
*********************



class MENU

{

public :

    void main_menue(void) ;

    void admin(void) ;

    void membermenue(void) ;

    void intro(void) ;


private :

    void edit_book(void);

    void edit_member(void) ;


} ;
```

```
//********************************************************
*********************

// THIS CLASS CONTAINS FUNCTIONS RELATED TO BOOKS

//********************************************************
*********************



class BOOK


{

public :

    char *bookaddr(int) ;

    char *bookname(int) ;

    void display(int) ;

    void list(void) ;



protected :

    int book_found(int) ;

    int bookname_found(char []) ;

    int recordno(int) ;

    int bookcodeof(char[]) ;
    int reccount(void) ;

    char *authorname(int) ;

    char *bookpublisher(int) ;

    void add_new_book(int, char tname[33], char tauthor[26],
char tpub[33], char taddr[50]) ;

    void identify_book(int) ;

    void modify(int, char[], char[], char[],char[]) ; void
deletion(void) ;

    void delete_rec(int) ;


private :

    int bookcode ;

    char name[33], author[26] ,pub[33] ,addr[50] ;

} ;



//******************************************************
********************

// THIS CLASS CONTAINS FUNCTIONS RELATED TO MEMBERS

//******************************************************
********************
```

```
class MEMBER
{
public :
    char *membername(int) ;
    void list(void) ;

protected :
    int member_found(int) ;
    int lastcode(void) ;
    char *memberphone(int) ;
    char *memberaddress(int) ;
    void add_mem(int, char [], char [], char[]) ;
    void modify(int, char[], char[], char[]) ;
    void update_book(int, int) ;
    void display(int) ;
    void delete_rec(int) ;

private :
    int memcode, bookcode ;
    char name[26], phone[10], address[33] ;
```

```
} ;



//**********************************************************
**********************

// THIS IS DERIVED FROM CLASS BOOK & MEMBER AND
CONTAINS FUNCTIONS FOR WORKING.

//**********************************************************
**********************



class WORKING : public BOOK, public MEMBER

{

public :

    void downloadbook(void) ;

    void add_book(void) ;

    void add_member(void) ;

    void modify_book(void) ;

    void modify_member(void) ;

    void delete_book(void) ;

    void delete_member(void) ;

    void searchbook(void) ;
```

```
} ;



//******************************************************
********************

// THIS CLASS CONTAINS FUNCTIONS RELATED DATE

//******************************************************
********************



class DATE

{

public :

    int day, mon, year ;


} ;



//******************************************************
********************

// FUNCTION FOR PASSWORD TO GO TO ADIMINSTRATIVE
MENU.
```

```
//***********************************************************
*********************

void password()
{
  clrscr();
  int ctr=0;
  int o;
  char pass[5];
  gotoxy(30,11);
  cout<<"ENTER PASSWORD : ";
  for(int i=0; i<5 && (o=getch())!=13 ; i++)
  {
   pass[i]=o;
   putch('*');
  }
  pass[i]='\0';
  ctr=strcmp(pass,"qwert");
  if(ctr==0)
  {
   MENU M;
   M.admin();
  }
  else
  {
   textcolor(RED+BLINK) ;
   gotoxy(32,13) ;
   cprintf("WRONG PASSWORD") ;
   textcolor(LIGHTGRAY) ;
   getch() ;
   return ;
  }
}


//**************************************************************************
// FUNCTION TO DISPLAY ADMINSTRATIVE MAIN MENU & CONTROL ALL THE FUNCTION MENU.
//**************************************************************************
```

```
void MENU :: admin(void)
{
  char ch ;
  while (1)
  { clrscr();
    colorbox(4,4,77,20);
    gotoxy(23,6) ;
    cout <<"A D M I N S T R A T I V E M E N U E" ;
    gotoxy(23,7) ;
    cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;
    gotoxy(32,10) ;
    cout <<"1. INTRODUCTION" ;
    gotoxy(32,11) ;
    cout <<"2. ADD NEW BOOK" ;
    gotoxy(32,12) ;
    cout <<"3. LIST OF BOOKS" ;
    gotoxy(32,13) ;
    cout <<"4. LIST OF MEMBERS" ;
    gotoxy(32,14) ;
    cout <<"5. EDIT BOOK" ;
    gotoxy(32,15) ;
```

```
cout <<"0. QUIT" ;

gotoxy(30,18) ;


cout <<"Enter your choice : " ;

ch = getche() ;

if(ch == 27)

  break ;

else if(ch == '1')

  intro() ;

else if (ch == '2')

{

  WORKING W ;

  W.add_book() ;

}

else if (ch == '3')

{

  BOOK B ;

  B.list() ;

}

else if (ch == '4')

{
      MEMBER M ;
      M.list() ;
    }
    else if (ch == '5')
    {
      MENU M;
      M.edit_book() ;
    }
    else if (ch == '0')
      break ;
  }
}


//********************************************************
*********************
// FUNCTION TO DISPLAY MAIN MENU & CONTROL ALL THE
FUNCTION MENU.
//********************************************************
*********************
```

```
void MENU :: main_menue(void)

{
  char ch ;
  while (1)
  { clrscr();
    colorbox(4,4,77,20);
    gotoxy(23,7) ;
    cout <<"O N L I N E B O O K L I B R A R Y";
    gotoxy(23,8) ;
    cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;
    gotoxy(32,10) ;
    cout <<"1. INDRODUCTION " ;
    gotoxy(32,11) ;
    cout <<"2. FOR MEMBERS" ;
    gotoxy(32,12) ;
    cout <<"3. FOR ADMINSTRATIVE" ;
    gotoxy(32,13) ;
    cout <<"4. SIGN UP ";
    gotoxy(32,14) ;
    cout <<"0. EXIT";
```

```
gotoxy(30,17);
cout <<"Enter your choice : " ;
ch = getche() ;
if(ch == 27)
  break ;
else if (ch =='1')
{ MENU M;
  M.intro() ;
}
else if (ch == '2')
{ MENU M;
  M.membermenue() ;
}
else if (ch =='3')
{
  password();
}
else if (ch=='4')
{
  WORKING W;
  W.add_member();
```

```
    }
    else if (ch == '0')
      break ;
  }
}


//******************************************************
********************
// FUNCTION TO DISPLAY MEMBER MENUE & CONTROL ALL
THE FUNCTION MENU.
//******************************************************
********************



void MENU :: membermenue(void)
{
  char ch ;
  while (1)
  { clrscr();
    colorbox(4,4,77,20);
    gotoxy(29,7) ;
```

```
cout <<"M E M B E R M E N U E";

gotoxy(29,8) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(32,10) ;

cout <<"1. LIST OF BOOKS" ;

gotoxy(32,11) ;

cout <<"2. SEARCH A BOOK" ;

gotoxy(32,12);

cout <<"3. DOWNLOAD";

gotoxy(32,13);

cout <<"4. EDIT USER";

gotoxy(32,14) ;

cout <<"0. EXIT" ;

gotoxy(30,17) ;

cout <<"Enter your choice : " ;

ch = getche() ;

if(ch == 27)

  break ;

else if (ch == '1')

{

  BOOK B ;
```

```
      B.list();
    }
    else if (ch == '2')
    {
      WORKING W ;
      W.searchbook();
    }
    else if (ch=='3')
    { WORKING W;
      W.downloadbook();
    }
    else if (ch=='4')
    {
      MENU W;
      W.edit_member();
    }
    else if (ch == '0')
      break ;
  }
}
```

```
//******************************************************
********************

// FUNCTION TO DISPLAY EDIT MENU FOR BOOK & CONTROL
FUNCTION IN THE EDIT MENU.

//******************************************************
********************



void MENU :: edit_book(void)

{

  char ch ;

  while (1)

  { clrscr() ;

    colorbox(4,4,77,20);

    gotoxy(31,7) ;

    cout <<"E D I T B O O K S";

    gotoxy(31,8) ;

    cout <<"~~~~~~~~~~~~~~~~~~" ;

    gotoxy(35,11) ;

    cout <<"1. MODIFY" ;

    gotoxy(35,12) ;
```

```
  cout <<"2. DELETE" ;
  gotoxy(35,13) ;
  cout <<"0. EXIT" ;
  gotoxy(32,16) ;
  cout <<"Enter your choice : " ;
  ch = getche() ;
  if(ch == 27)
    break ;
  else if (ch == '1')
  {
    WORKING W ;
    W.modify_book() ;
  }
  else if (ch == '2')
  {
    WORKING W ;
    W.delete_book() ;
  }
  else if (ch == '0')
    break ;
}
```

```
}



//********************************************************
*********************

// FUNCTION TO DISPLAY EDIT MENU FOR MEMBERS &
CONTROL FUNCTION IN THE EDIT MENU.

//********************************************************
*********************



void MENU :: edit_member(void)

{

  char ch ;

  while (1)

  {

    clrscr() ;

    colorbox(4,4,77,20);

    gotoxy(30,7) ;

    cout <<"E D I T M E M B E R S";

    gotoxy(30,8) ;

    cout <<"~~~~~~~~~~~~~~~~~~~~~~" ;
```

```
gotoxy(35,11) ;
cout <<"1. MODIFY" ;
gotoxy(35,12) ;
cout <<"2. DELETE" ;
gotoxy(35,13) ;
cout <<"0. EXIT" ;
gotoxy(32,16) ;
cout <<"Enter your choice : " ;
ch = getche() ;
if(ch == 27)
  break ;
else if (ch == '1')
{
  WORKING W ;
  W.modify_member() ;
}
else if (ch == '2')
{
  WORKING W ;
  W.delete_member() ;
}
```

```
    else if (ch == '0')
      break ;
  }
}

//********************************************************
*********************
// FUNCTION TO DISPLAY THE INTRODUCTION OF THE
PROJECT.
//********************************************************
*********************



void MENU :: intro(void)
{
  clrscr() ;
  gotoxy(30,5) ;
  cout <<"Welcome to Project" ;
  textcolor(BLACK+BLINK) ;
  textbackground(WHITE) ;
  gotoxy(27,7) ;
  cprintf(" ONLINE BOOK LIBRARY ") ;
  textcolor(LIGHTGRAY) ;

  textbackground(BLACK) ;

  gotoxy(15,10) ;

  cout <<"This project has facility of maintaining records" ;

  gotoxy(15,11) ;

  cout <<"of BOOKS and MEMBERS." ;

  gotoxy(15,14) ;

  cout <<"This project can hold more than 10,000 books" ;

  gotoxy(15,15) ;

  cout <<"records." ;

  gotoxy(15,18) ;

  cout <<"One member can download one book at a time." ;

  textcolor(LIGHTGRAY+BLINK) ;

  gotoxy(27,24) ;

  cprintf("Press any key to continue") ;

  textcolor(LIGHTGRAY) ;

  getch() ;

}



//******************************************************
********************
```

```
// THIS FUNCTION RETURNS THE NO. OF RECORDS IN BOOK
FILE.

//********************************************************
*********************


int BOOK :: reccount(void)

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  file.seekg(0,ios::beg) ;

  int count=0 ;

  while (file.read((char *) this, sizeof(BOOK)))

   count++ ;

  file.close() ;

  return count ;

}



//********************************************************
*********************

// THIS FUNCTION RETURN 0 IF GIVEN BOOK CODE NOT FOUND
```

```
//********************************************************
*********************



int BOOK :: book_found(int tcode)

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  file.seekg(0,ios::beg) ;

  int found=0 ;

  while (file.read((char *) this, sizeof(BOOK)))

  {

    if (bookcode == tcode)

    {

      found = 1 ;

      break ;

    }

  }

  file.close() ;

  return found ;

}
```

```
//*****************************************************
********************

// THIS FUNCTION RETURN 0 IF GIVEN BOOK NAME NOT
FOUND

//*****************************************************
********************



int BOOK :: bookname_found(char t1code[33])

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  file.seekg(0,ios::beg) ;

  int found=0 ;

  while (file.read((char *) this, sizeof(BOOK)))

  {

    if (!strcmpi(name,t1code))

    {

      found = 1 ;

      break ;
    }
  }
  file.close() ;
  return found ;
}



//*********************************************************
*********************
// THIS FUNCTION RETURNS THE BOOK NAME OF THE GIVEN
BOOK CODE.
//*********************************************************
*********************



char *BOOK :: bookname(int tcode)
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  file.seekg(0,ios::beg) ;
  char tname[33] ;
  while (file.read((char *) this, sizeof(BOOK)))
  {
    if (bookcode == tcode)
    {
      strcpy(tname,name) ;
      break ;
    }
  }
  file.close() ;
  return tname ;
}


//********************************************************
*********************
// THIS FUNCTION RETURNS THE AUTHOR NAME OF THE GIVEN
BOOK CODE.
//********************************************************
*********************


char *BOOK :: authorname(int tcode)
```

```
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  file.seekg(0,ios::beg) ;
  char tauthor[26] ;
  while (file.read((char *) this, sizeof(BOOK)))
  {
    if (bookcode == tcode)
    {
      strcpy(tauthor,author) ;
      break ;
    }
  }
  file.close() ;
  return tauthor ;
}


//********************************************************
*********************
```

```
// THIS FUNCTION RETURNS THE ADDRESS OF THE GIVEN
BOOK CODE.

//******************************************************
********************


char *BOOK :: bookaddr(int tcode)

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  file.seekg(0,ios::beg) ;

  char taddr[50] ;

  while (file.read((char *) this, sizeof(BOOK)))

  {

    if (bookcode == tcode)

    {

      strcpy(taddr,addr) ;

      break ;

    }

  }

  file.close() ;

  return taddr ;
```

```
}



//*********************************************************
*********************

// THIS FUNCTION RETURNS THE PUBLISHER NAME OF THE
GIVEN BOOK CODE.

//*********************************************************
*********************



char *BOOK :: bookpublisher(int tcode)

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  file.seekg(0,ios::beg) ;

  char tpub[33] ;

  while (file.read((char *) this, sizeof(BOOK)))

  {

    if (bookcode == tcode)

    {
      strcpy(tpub,pub) ;
      break ;
    }
  }
  file.close() ;
  return tpub ;
}



//********************************************************
*********************
// THIS FUNCTION RETURNS THE BOOK CODE OF THE GIVEN
BOOK NAME.
//********************************************************
*********************



int BOOK :: bookcodeof(char t1code[33])
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  file.seekg(0,ios::beg) ;
  int tcode=0 ;
  while (file.read((char *) this, sizeof(BOOK)))
  {
    if (!strcmpi(name,t1code))
    {
      tcode = bookcode ;
      break ;
    }
  }
  file.close() ;
  return tcode ;
}



//********************************************************
*********************
// THIS FUNCTION DELETES THE RECORD OF THE GIVEN BOOK
CODE.
//********************************************************
*********************
```

```
void BOOK :: delete_rec(int tcode)
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  fstream temp ;
  temp.open("TEMP.DAT", ios::out) ;
  system("attrib +H TEMP.DAT");
  file.seekg(0,ios::beg) ;
  while ( !file.eof() )
  {
    file.read((char *) this, sizeof(BOOK)) ;
    if ( file.eof() )
      break ;
    if ( bookcode != tcode )
      temp.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
  file.open("BOOK.DAT", ios::out) ;
  system("attrib +H BOOK.DAT");
  temp.open("TEMP.DAT", ios::in) ;
  system("attrib +H TEMP.DAT");
  temp.seekg(0,ios::beg) ;
  while ( !temp.eof() )
  {
    temp.read((char *) this, sizeof(BOOK)) ;
    if ( temp.eof() )
      break ;
    file.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
}


//********************************************************
*********************
// THIS FUNCTION ADD THE RECORD IN THE BOOK FILE
//********************************************************
*********************
```

```
void BOOK :: add_new_book(int tcode,char tname[33], char
tauthor[26], char tpub[33] ,char taddr[50])

{

  fstream file ;

  file.open("BOOK.DAT", ios::app) ;

  system("attrib +H BOOK.DAT");

  bookcode = tcode ;

  strcpy(name,tname) ;

  strcpy(author,tauthor) ;

  strcpy(pub,tpub) ;

  strcpy(addr,taddr) ;

  file.write((char *) this, sizeof(BOOK)) ;

  file.close() ;

}



//*****************************************************
********************

// THIS FUNCTION SHOW THE RECORD IN THE BOOK FILE FOR
THE GIVEN BOOK CODE.

//*****************************************************
********************
```

```
void BOOK :: identify_book(int tcode)
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  fstream temp ;
  temp.open("TEMP.DAT", ios::out) ;
  system("attrib +H TEMP.DAT");
  file.seekg(0,ios::beg) ;
  while ( !file.eof() )
  {
    file.read((char *) this, sizeof(BOOK)) ;
    if ( file.eof() )
      break ;
    if ( bookcode == tcode )
    {
      temp.write((char *) this, sizeof(BOOK)) ;
    }
    else
      temp.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
  file.open("BOOK.DAT", ios::out) ;
  system("attrib +H BOOK.DAT");
  temp.open("TEMP.DAT", ios::in) ;
  system("attrib +H TEMP.DAT");
  temp.seekg(0,ios::beg) ;
  while ( !temp.eof() )
  {
    temp.read((char *) this, sizeof(BOOK)) ;
    if ( temp.eof() )
      break ;
    file.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
}


//***********************************************************
*********************
```

```
// THIS FUNCTION MODIFY THE RECORD IN THE BOOK FILE
FOR THE GIVEN BOOK CODE.

//********************************************************
*********************



void BOOK :: modify(int tcode, char tname[33], char tauthor[26],
char taddr[50] ,char tpub[50] )

{

  fstream file ;

  file.open("BOOK.DAT", ios::in) ;

  system("attrib +H BOOK.DAT");

  fstream temp ;

  temp.open("TEMP.DAT", ios::out) ;

  system("attrib +H TEMP.DAT");

  file.seekg(0,ios::beg) ;

  while ( !file.eof() )

  {

    file.read((char *) this, sizeof(BOOK)) ;

    if ( file.eof() )

      break ;

    if ( bookcode == tcode )
    {
      strcpy(name,tname) ;
      strcpy(author,tauthor) ;
      strcpy(addr,taddr) ;
      strcpy(pub,tpub) ;
      temp.write((char *) this, sizeof(BOOK)) ;
    }
    else
      temp.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
  file.open("BOOK.DAT", ios::out) ;
  system("attrib +H BOOK.DAT");
  temp.open("TEMP.DAT", ios::in) ;
  system("attrib +H TEMP.DAT");
  temp.seekg(0,ios::beg) ;
  while ( !temp.eof() )
  {
    temp.read((char *) this, sizeof(BOOK)) ;
    if ( temp.eof() )
```

```
      break ;
    file.write((char *) this, sizeof(BOOK)) ;
  }
  file.close() ;
  temp.close() ;
}



//*******************************************************
*********************
// THIS FUNCTION DISPLAY THE RECORD FROM THE BOOK FILE
FOR THE GIVEN BOOK CODE.
//*******************************************************
*********************



void BOOK :: display(int tcode)
{
  fstream file ;
  file.open("BOOK.DAT", ios::in) ;
  system("attrib +H BOOK.DAT");
  file.seekg(0,ios::beg) ;
  while (file.read((char *) this, sizeof(BOOK)))
  {
    if (bookcode == tcode)
    {
      gotoxy(5,5) ;
      cout <<"Book Code : " <<bookcode ;
      gotoxy(5,7) ;
      cout <<"Book Name : " <<name ;
      gotoxy(5,8) ;
      cout <<"Author Name : " <<author ;
      gotoxy(5,9) ;
      cout <<"Publisher : " <<pub ;
      gotoxy(5,10) ;
      cout <<"Address : " <<addr ;
      break ;
    }
  }
  file.close() ;
}
```

```
//************************************************************************
// THIS FUNCTION DISPLAY THE LIST OF BOOKS.
//************************************************************************



void BOOK :: list(void)
{
  clrscr() ;
  int row = 6 , found=0, flag=0 ;
  char ch ;
  gotoxy(33,2) ;
  cout <<"LIST OF BOOKS" ;
  gotoxy(32,3) ;
  cout <<"~~~~~~~~~~~~~~~~" ;
  gotoxy(1,4) ;
  cout <<"CODE    BOOK NAME        AUTHOR      PUBLISHER ADDRESS" ;
  gotoxy(1,5) ;
  cout <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~" ;
```

```
fstream file ;
file.open("BOOK.DAT", ios::in) ;
system("attrib +H BOOK.DAT");
file.seekg(0,ios::beg) ;
while (file.read((char *) this, sizeof(BOOK)))
{
 flag = 0 ;
 delay(100) ;
 found = 1 ;
 gotoxy(2,row) ;
 cout <<bookcode ;
 gotoxy(10,row) ;
 cout <<name ;
 gotoxy(25,row) ;
 cout <<author ;
 gotoxy(39,row) ;
 cout <<pub ;
 gotoxy(57,row) ;
 cout <<addr ;
 if ( row == 22 )
 {
    flag = 1 ;
    row = 6 ;
    gotoxy(1,25) ;
    cout <<"Press any key to continue or Press <ESC> to exit" ;
    ch = getch() ;
    if (ch == 27)
      break ;
    clrscr() ;
    gotoxy(33,2) ;
    cout <<"LIST OF BOOKS" ;
    gotoxy(32,3) ;
    cout <<"~~~~~~~~~~~~~~~~~" ;
    gotoxy(1,4) ;
    cout <<"CODE    BOOK NAME    AUTHOR   PUBLISHER" ;
    gotoxy(1,5) ;
    cout
<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~" ;
   }
   else
   row = row + 2 ;
  }
```

```
  if (!found)
  {
    gotoxy(5,10) ;
    cout <<"\7Records not found" ;
  }
  if (!flag)
  {
    gotoxy(1,25) ;
    cout <<"Press any key to continue..." ;
    getche() ;
  }
  file.close () ;
}


//*********************************************************
*********************
// THIS FUNCTION RETURN 0 IF THE GIVEN MEMBER CODE
NOT FOUND.
//*********************************************************
*********************
```

```
int MEMBER :: member_found(int mcode)
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  file.seekg(0,ios::beg) ;
  int found=0 ;
  while (file.read((char *) this, sizeof(MEMBER)))
  {
    if (memcode == mcode)
    {
      found = 1 ;
      break ;
    }
  }
  file.close() ;
  return found ;
}

//********************************************************
*********************
```

```
// THIS FUNCTION RETURN THE LAST CODE OF THE MEMBER
FILE.

//*****************************************************
********************



int MEMBER :: lastcode(void)

{

  fstream file ;

  file.open("MEMBER.DAT", ios::in) ;

  system("attrib +H MEMBER.DAT");

  file.seekg(0,ios::beg) ;

  int mcode=0 ;

  while (file.read((char *) this, sizeof(MEMBER)))

    mcode = memcode ;

  file.close() ;

  return mcode ;

}


//*****************************************************
********************

// THIS FUNCTION RETURNS MEMBER NAME OF THE GIVEN
MEMBER CODE.
```

```
//*******************************************************
*********************



char *MEMBER :: membername(int mcode)
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  file.seekg(0,ios::beg) ;
  char mname[26] ;
  while (file.read((char *) this, sizeof(MEMBER)))
  {
    if (memcode == mcode)
    {
      strcpy(mname,name) ;
      break ;
    }
  }
  file.close() ;
  return mname ;
}
```

```
//**********************************************************
**********************

// THIS FUNCTION RETURNS MEMBER PHONE NUMBER OF THE
GIVEN MEMBER CODE.

//**********************************************************
**********************



char *MEMBER :: memberphone(int mcode)
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  file.seekg(0,ios::beg) ;
  char mphone[10] ;
  while (file.read((char *) this, sizeof(MEMBER)))
  {
    if (memcode == mcode)
    {
      strcpy(mphone,phone) ;
      break ;
    }
  }
  file.close() ;
  return mphone ;
}



//***********************************************************
*********************

// THIS FUNCTION RETURNS MEMBER ADDRESS OF THE GIVEN
MEMBER CODE.

//***********************************************************
*********************



char *MEMBER :: memberaddress(int mcode)
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  file.seekg(0,ios::beg) ;
  char maddress[33] ;
  while (file.read((char *) this, sizeof(MEMBER)))
  {
    if (memcode == mcode)
    {
      strcpy(maddress,address) ;
      break ;
    }
  }
  file.close() ;
  return maddress ;
}


//******************************************************
********************
// THIS FUNCTION DELETE RECORD FOR THE GIVEN MEMBER
CODE.
//******************************************************
********************


void MEMBER :: delete_rec(int mcode)
```

```
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  fstream temp ;
  temp.open("TEMP.DAT", ios::out) ;
  system("attrib +H TEMP.DAT");
  file.seekg(0,ios::beg) ;
  while ( !file.eof() )
  {
    file.read((char *) this, sizeof(MEMBER)) ;
    if ( file.eof() )
      break ;
    if ( memcode != mcode )
      temp.write((char *) this, sizeof(MEMBER)) ;
  }
  file.close() ;
  temp.close() ;
  file.open("MEMBER.DAT", ios::out) ;
  system("attrib +H MEMBER.DAT");
  temp.open("TEMP.DAT", ios::in) ;
  system("attrib +H TEMP.DAT");
  temp.seekg(0,ios::beg) ;
  while ( !temp.eof() )
  {
    temp.read((char *) this, sizeof(MEMBER)) ;
    if ( temp.eof() )
      break ;
    file.write((char *) this, sizeof(MEMBER)) ;
  }
  file.close() ;
  temp.close() ;
}

//******************************************************
********************
// THIS FUNCTION UPDATE RECORD FOR THE GIVEN MEMBER
CODE.
//******************************************************
********************



void MEMBER :: update_book(int mcode, int tcode)
```

```
{
  fstream file ;
  file.open("MEMBER.DAT", ios::in) ;
  system("attrib +H MEMBER.DAT");
  fstream temp ;
  temp.open("TEMP.DAT", ios::out) ;
  system("attrib +H TEMP.DAT");
  file.seekg(0,ios::beg) ;
  while ( !file.eof() )
  {
    file.read((char *) this, sizeof(MEMBER)) ;
    if ( file.eof() )
      break ;
    if ( memcode == mcode )
    {
      bookcode = tcode ;
      temp.write((char *) this, sizeof(MEMBER)) ;
    }
    else
      temp.write((char *) this, sizeof(MEMBER)) ;
  }
  file.close() ;
  temp.close() ;
  file.open("MEMBER.DAT", ios::out) ;
  system("attrib +H MEMBER.DAT");
  temp.open("TEMP.DAT", ios::in) ;
  system("attrib +H TEMP.DAT");
  temp.seekg(0,ios::beg) ;
  while ( !temp.eof() )
  {
    temp.read((char *) this, sizeof(MEMBER)) ;
    if ( temp.eof() )
      break ;
    file.write((char *) this, sizeof(MEMBER)) ;
  }
  file.close() ;
  temp.close() ;
}


//********************************************************
*********************
```

```
// THIS FUNCTION MODIFY RECORD FOR THE GIVEN MEMBER CODE.

//*****************************************************************************



void MEMBER :: modify(int mcode, char mname[26], char mphone[10], char maddress[33])

{

 fstream file ;

 file.open("MEMBER.DAT", ios::in) ;

 system("attrib +H MEMBER.DAT");

 fstream temp ;

 temp.open("TEMP.DAT", ios::out) ;

 system("attrib +H TEMP.DAT");

 file.seekg(0,ios::beg) ;

 while ( !file.eof() )

 {

  file.read((char *) this, sizeof(MEMBER)) ;

  if ( file.eof() )

   break ;

  if ( memcode == mcode )
  {
    strcpy(name,mname) ;
    strcpy(phone,mphone) ;
    strcpy(address,maddress) ;
    temp.write((char *) this, sizeof(MEMBER)) ;
  }
 else
    temp.write((char *) this, sizeof(MEMBER)) ;
}
file.close() ;
temp.close() ;
file.open("MEMBER.DAT", ios::out) ;
system("attrib +H MEMBER.DAT");
temp.open("TEMP.DAT", ios::in) ;
system("attrib +H TEMP.DAT");
temp.seekg(0,ios::beg) ;
while ( !temp.eof() )
{
  temp.read((char *) this, sizeof(MEMBER)) ;
  if ( temp.eof() )
    break ;
    file.write((char *) this, sizeof(MEMBER)) ;
  }
  file.close() ;
  temp.close() ;
}



//*****************************************************************************
// THIS FUNCTION ADD RECORD IN THE FILE FOR THE GIVEN MEMBER CODE.
//*****************************************************************************



void MEMBER :: add_mem(int mcode, char mname[26], char maddress[33], char mphone[10])
{
  fstream file ;
  file.open("MEMBER.DAT", ios::app) ;
  system("attrib +H MEMBER.DAT");
  memcode = mcode ;
  strcpy(name,mname) ;
  strcpy(phone,mphone) ;

  strcpy(address,maddress) ;

  file.write((char *) this, sizeof(MEMBER)) ;

  file.close() ;

}



//******************************************************
********************

// THIS FUNCTION DISPLAY THE RECORD FOR THE GIVEN
MEMBER CODE.

//******************************************************
********************



void MEMBER :: display(int mcode)

{

  fstream file ;

  file.open("MEMBER.DAT", ios::in) ;

  system("attrib +H MEMBER.DAT");

  file.seekg(0,ios::beg) ;

  while (file.read((char *) this, sizeof(MEMBER)))
  {
    if (memcode == mcode)
    {
      gotoxy(5,3) ;
      cout <<"Member Code # " <<mcode ;
      gotoxy(5,4) ;
      cout <<"~~~~~~~~~~~~~~~~~" ;
      gotoxy(5,6) ;
      cout <<"Name :" <<name;
      gotoxy(5,7) ;
      cout<<"Phone :" <<phone;
      gotoxy(5,8);
      cout <<"Address : " <<address ;
      break ;
    }
  }
  file.close() ;
}


//********************************************************
*********************
```

```
// THIS FUNCTION DISPLAY THE LIST OF THE MEMBERS

//**********************************************************
**********************



void MEMBER :: list(void)

{

  clrscr() ;


  BOOK B ;

  int row = 6 , found=0, flag=0 ;

  char ch ;

  gotoxy(32,2) ;

  cout <<"LIST OF MEMBERS" ;

  gotoxy(31,3) ;

  cout <<"~~~~~~~~~~~~~~~~~~~" ;

  gotoxy(1,4) ;

  cout <<"CODE               NAME                    PHONE      " ;

  gotoxy(1,5) ;

  cout
<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~" ;
```

```
fstream file ;
file.open("MEMBER.DAT", ios::in) ;
system("attrib +H MEMBER.DAT");
file.seekg(0,ios::beg) ;
while (file.read((char *) this, sizeof(MEMBER)))
{
  flag = 0 ;
  delay(20) ;
  found = 1 ;
  gotoxy(2,row) ;
  cout <<memcode ;
  gotoxy(10,row) ;
  cout <<name ;
  gotoxy(35,row) ;
  cout <<phone ;
  if ( row == 22 )
  {
    flag = 1 ;
    row = 6 ;
    gotoxy(1,25) ;
    cout <<"Press any key to continue or Press <ESC> to exit" ;
    ch = getch() ;
    if (ch == 27)
     break ;
    clrscr() ;
    gotoxy(32,2) ;
    cout <<"LIST OF MEMBERS" ;
    gotoxy(31,3) ;
    cout <<"~~~~~~~~~~~~~~~~~" ;
    gotoxy(1,4) ;
    cout <<"CODE BOOK CODE NAME     PHONE" ;
    gotoxy(1,5) ;
    cout
 <<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~" ;
   }
   else
    row = row + 2 ;
  }
  if (!found)
  {
   gotoxy(5,10) ;
   cout <<"\n\tRecords not found" ;
```

```
  }
  if (!flag)
  {
    gotoxy(1,25) ;
    cout <<"Press any key to continue..." ;
    getche() ;
  }
  file.close () ;
}


//********************************************************
*********************
// FUNCTION TO DISPLAY THE BOOK FOR THE GIVEN
BOOKCODE.
//********************************************************
*********************



void WORKING :: searchbook(void)
{
  char t1code[5];
  char ch ;
  int t2code=0, tcode=0 ;
  int valid ;
  do
  {
    valid = 1 ;
    while (1)
    {
      clrscr() ;
      gotoxy(72,1) ;
      cout <<"<0>=Exit" ;
      gotoxy(5,5) ;
      cout <<"Enter Code of the Book to be Search:" ;
      gotoxy(5,6) ;
      cout <<"    or     " ;
      gotoxy(5,7) ;
      cout <<"Press <ENTER> for help " ;
      gets(t1code) ;
      if (t1code[0] == '0')
        return ;
      if (strlen(t1code) == 0)
```

```
      {
        BOOK B;
        B.list() ;
      }
      else
        break ;
    }
    t2code = atoi(t1code) ;
    tcode = t2code ;
    if ((tcode == 0 && !bookname_found(t1code)) || (tcode != 0 && !book_found(tcode)))
    {
      valid = 0 ;
      gotoxy(5,10) ;
      cout <<"\7Record not found" ;
      gotoxy(5,11) ;
      cout <<"Press <ESC> to exit or any other key to continue..." ;
      ch = getch() ;
      if (ch == 27)
        return ;
    }
  } while (!valid) ;
  if (tcode == 0)
    tcode = bookcodeof(t1code) ;
  clrscr() ;
  gotoxy(72,1) ;
  cout <<"<0>=Exit" ;
  BOOK::display(tcode) ;
  getch();
}


//*********************************************************
*********************
// THIS FUNCTION TO ADD RECORD IN THE BOOK FILE.
//*********************************************************
*********************



void WORKING :: add_book(void)
{
  if (!reccount())
  {
    add_new_book(0,"null","null","null","null") ;
    BOOK::delete_rec(0) ;
  }
  char ch ;
  int tcode ;
  char tname[33], tauthor[26] ,tpubl[33] ,taddr[50] ;
  do
  {
    int found=0, valid=0 ; int tc ;
    char t[10], t1[10] ;
    clrscr() ;
    gotoxy(29,3) ;
    cout <<"ADDITION OF THE BOOKS" ;
    gotoxy(29,4) ;
    cout <<"~~~~~~~~~~~~~~~~~~~~~" ;
    gotoxy(72,1) ;
    cout <<"<0>=Exit" ;
    gotoxy(5,25) ;
    cout <<"Enter code no. of the book" ;
    gotoxy(5,5) ;
    cout <<"Code no. " ;
    gets(t) ;
```

```
tc = atoi(t) ;
tcode = tc ;
if (tcode == 0)
  return ;
if (book_found(tcode))
{
  found = 1 ;
  gotoxy(19,8) ;
  cout <<bookname(tcode) ;
  gotoxy(19,9) ;
  cout <<authorname(tcode) ;
  gotoxy(19,10) ;
  cout <<bookpublisher(tcode) ;
  gotoxy(19,11);
  cout <<bookaddr(tcode);
}
gotoxy(5,8) ;
cout <<"Book Name : " ;
gotoxy(5,9) ;
cout <<"Author Name : " ;
gotoxy(5,10) ;
```

```
cout <<"Publisher : " ;
gotoxy(5,11) ;
cout <<"Address : " ;
valid = 0 ;
while (!valid && !found)
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the name of the book" ;
  gotoxy(19,8) ;
  clreol() ;
  gets(tname) ;
  strupr(tname) ;
  if (tname[0] == '0')
    return ;
  if (strlen(tname) < 1 || strlen(tname) > 32)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout<<"\nEnter correctly (Range: 1..32)" ;
    getch() ;
  }
}
valid = 0 ;
while (!valid && !found)
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the author's name of the book" ;
  gotoxy(19,9) ;
  clreol() ;
  gets(tauthor) ;
  strupr(tauthor) ;
  if (tauthor[0] == '0')
    return ;
  if (strlen(tauthor) < 1 || strlen(tauthor) > 25)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\7Enter correctly (Range: 1..25)" ;
    getch() ;
  }
}
valid = 0 ;
while (!valid && !found)
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the publisher of the book" ;
  gotoxy(19,10) ;
  clreol() ;
  gets(tpubl) ;
  strupr(tpubl) ;
  if (tpubl[0] == '0')
    return ;
  if (strlen(tpubl) < 1 || strlen(tpubl) > 32)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\nEnter correctly (Range: 1..32)" ;
    getch() ;
  }
}
valid = 0 ;
while (!valid && !found)
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the address of the book" ;
  gotoxy(19,11) ;
  clreol() ;
  gets(taddr) ;
  strupr(taddr) ;
  if (tname[0] == '0')
    return ;
  if (strlen(taddr) < 1 || strlen(taddr) > 32)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\nEnter correctly (Range: 1..32)" ;
    getch() ;
  }
}
gotoxy(5,25) ;
clreol() ;
do
{
  gotoxy(5,15) ;
  clreol() ;
  cout <<"Do you want to save (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
}while (ch != 'Y' && ch != 'N') ;
if (ch == 'Y')
{
  if (found)
    identify_book(tcode) ;
```

```
      else
        add_new_book(tcode,tname,tauthor,tpubl,taddr) ;
    }
    do
    {
      gotoxy(5,17) ;
      clreol() ;
      cout <<"Do you want to add more (y/n) : " ;
      ch = getche() ;
      ch = toupper(ch) ;
    } while (ch != 'Y' && ch != 'N') ;
  } while (ch == 'Y') ;
}



//********************************************************
*********************

// THIS DATA TO ADD RECORD IN THE MEMBER FILE

//********************************************************
*********************
```

```
void WORKING :: add_member(void)
{
 char ch ;
 int mcode ;
 char mname[26], mphone[10], maddress[33] ;
 mcode = lastcode() ;
 if (mcode == 0)
 {
  add_mem(mcode,"null","null","null") ;
  MEMBER::delete_rec(mcode) ;
 }
 mcode++ ;
 do
 {
  int valid=0 ;
  clrscr() ;
  gotoxy(28,3) ;
  cout <<"ADDITION OF THE MEMBERS" ;
  gotoxy(28,4) ;
  cout <<"~~~~~~~~~~~~~~~~~~~~~~~~" ;
  gotoxy(72,1) ;
```

```
cout <<"<0>=Exit" ;
gotoxy(5,7) ;
cout <<"Member Code # " <<mcode ;
gotoxy(5,8) ;
cout <<"~~~~~~~~~~~~~~~~~" ;
gotoxy(5,10) ;
cout <<"Name : " ;
gotoxy(5,12) ;
cout <<"Phone : " ;
gotoxy(5,14) ;
cout <<"Address : " ;
do
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the name of the New Member" ;
  gotoxy(15,10) ;
  clreol() ;
  gets(mname) ;
  strupr(mname) ;
```

```
      if (mname[0] == '0')
        return ;
      if (strlen(mname) < 1 || strlen(mname) > 25)
      {
        valid = 0 ;
        gotoxy(5,25) ;
        clreol() ;
        cout <<"\7Enter correctly (Range: 1..25)" ;
        getch() ;
      }
    } while (!valid) ;
    do
    {
      valid = 1 ;
      gotoxy(5,25) ;
      clreol() ;
      cout <<"Enter Phone no. of the Member or Press <ENTER>
for none" ;
      gotoxy(15,12) ;
      clreol() ;
      gets(mphone) ;
      if (mphone[0] == '0')
```

```
        return ;
      if ((strlen(mphone) < 10 && strlen(mphone) > 0) || (strlen(mphone) > 12))
      {
        valid = 0 ;
        gotoxy(5,25) ;
        clreol() ;
        cout <<"\7Enter correctly" ;
        getch() ;
      }
    } while (!valid) ;
    if (strlen(mphone) == 0)
      strcpy(mphone,"-") ;
    do
    {
      valid = 1 ;
      gotoxy(5,25) ;
      clreol() ;
      cout <<"Enter the address of the New Member" ;
      gotoxy(15,14) ;
      clreol() ;
      gets(maddress) ;
```

```
  strupr(maddress) ;
  if (maddress[0] == '0')
    return ;
  if (strlen(maddress) < 1 || strlen(maddress) > 32)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\7Enter correctly (Range: 1..32)" ;
    getch() ;
  }
} while (!valid) ;
gotoxy(5,25) ;
clreol() ;
do
{
  gotoxy(5,17) ;
  clreol() ;
  cout <<"Do you want to save (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
```

```
      if (ch == '0')
        return ;
    } while (ch != 'Y' && ch != 'N') ;
    if (ch == 'Y')
    {
      add_mem(mcode,mname,maddress,mphone) ;
      mcode++ ;
    }
    do
    {
      gotoxy(5,19) ;
      clreol() ;
      cout <<"Do you want to add more (y/n) : " ;
      ch = getche() ;
      ch = toupper(ch) ;
      if (ch == '0')
        return ;
    } while (ch != 'Y' && ch != 'N') ;
  } while (ch == 'Y') ;
}
```

```
//******************************************************
********************

// THIS FUNCTION DOWNLOAD THE BOOK

//******************************************************
********************



void WORKING :: downloadbook(void)

{

  BOOK B ;

  MEMBER M ;

  DATE D ;

  char t1code[33], ch ;

  char sor[100],dest[100];

  int t2code=0, tcode=0, mcode=0 ;

  int valid ;

  int d1, m1, y1 ;

  struct date d;

  getdate(&d);

  d1 = d.da_day ;

  m1 = d.da_mon ;
```

```
y1 = d.da_year ;
do
{
  valid = 1 ;
  while (1)
  {
    clrscr() ;
    gotoxy(5,2) ;
    cout <<"Date : " <<d1 <<"/" <<m1 <<"/" <<y1 ;
    gotoxy(72,1) ;
    cout <<"<0>=Exit" ;
    gotoxy(5,5) ;
    cout <<"Enter Code or Name of the Book to be download" ;
    gotoxy(5,6) ;
    cout <<"     or     " ;
    gotoxy(5,7) ;
    cout <<"Press <ENTER> for help " ;
    gets(t1code) ;
    if (t1code[0] == '0')
      return ;
    if (strlen(t1code) == 0)
```

```
        B.list() ;
      else
        break ;
    }
    t2code = atoi(t1code) ;
    tcode = t2code ;
    if ((tcode == 0 && !bookname_found(t1code)) || (tcode != 0 && !book_found(tcode)))
    {
      valid = 0 ;
      gotoxy(5,10) ;
      cout <<"\7Record not found" ;
      gotoxy(5,11) ;
      cout <<"Press <ESC> to exit or any other key to continue..." ;
      ch = getch() ;
      if (ch == 27)
        return ;
    }
  } while (!valid) ;
  if (tcode == 0)
    tcode = bookcodeof(t1code) ;
  do
```

```
{
  valid = 1 ;
  while (1)
  {
    clrscr() ;
    gotoxy(72,1) ;
    cout <<"<0>=Exit" ;
    gotoxy (5,2) ;
    cout <<"Date : " <<d1 <<"/" <<m1 <<"/" <<y1 ;
    gotoxy(5,5) ;
    cout <<"Book Name: " <<bookname(tcode) ;
    gotoxy(5,7) ;
    cout <<"Enter Code no. of the Member" ;
    gotoxy(5,8) ;
    cout <<"    or    " ;
    gotoxy(5,9) ;
    cout <<"Press <ENTER> for help " ;
    gets(t1code) ;
    if (t1code[0] == '0')
      return ;
    if (strlen(t1code) == 0)
    M.list() ;
  else
    break ;
}
t2code = atoi(t1code) ;
mcode = t2code ;
if (mcode == 0)
{
  valid = 0 ;
  gotoxy(5,25) ;
  cout <<"\7Enter Correctly" ;
  getch() ;
}
if (!member_found(mcode) && valid)
{
  valid = 0 ;
  gotoxy(5,13) ;
  cout <<"\7Record not found" ;
  gotoxy(5,14) ;
  cout <<"Press <ESC> to exit or any other key to continue..." ;
  ch = getch() ;
```

```
      if (ch == 27)
        return ;
    }
  } while (!valid) ;
  clrscr();
  identify_book(tcode) ;
  update_book(mcode,tcode) ;
  B.display(tcode);
  storedate(); //TO STORE THE DATE OF DOWNLOAD
  strcpy(sor,B.bookaddr(tcode));
  strcpy(dest,B.bookname(tcode));
  link(sor,dest); //TO DOWNLOAD LINK OF BOOK gotoxy(5,13) ;
  cout <<"\7Book is downloading to " <<membername(mcode) ;
  textcolor(BLACK+BLINK) ;
  textbackground(WHITE) ;
  gotoxy(5,15) ;
  cprintf(" ! BOOK DOWNLOADED ! ") ;
  textcolor(LIGHTGRAY) ;
  textbackground(BLACK) ;
  getch();
}
```

```
//******************************************************
********************

// THIS FUNCTION TO MODIFY THE BOOK RECORD

//******************************************************
********************



void WORKING :: modify_book(void)

{

  BOOK B ;

  char t1code[5], tname[33], tauthor[26], ch ,taddr[50],tpub[33];

  int t2code=0, tcode=0 ;

  int valid ;

  do

  {

    valid = 1 ;

    while (1)

    {

      clrscr() ;

      gotoxy(72,1) ;
```

```
     cout <<"<0>=Exit" ;
     gotoxy(5,5) ;
     cout <<"Enter Code or Name of the Book to be modified" ;
     gotoxy(5,6) ;
     cout <<"    or    " ;
     gotoxy(5,7) ;
     cout <<"Press <ENTER> for help " ;
     gets(t1code) ;
     if (t1code[0] == '0')
       return ;
     if (strlen(t1code) == 0)
       B.list() ;
     else
       break ;
   }
   t2code = atoi(t1code) ;
   tcode = t2code ;
   if ((tcode == 0 && !bookname_found(t1code)) || (tcode != 0
&& !book_found(tcode)))
   {
     valid = 0 ;
     gotoxy(5,10) ;
```

```
    cout <<"\7Record not found" ;
    gotoxy(5,11) ;
    cout <<"Press <ESC> to exit or any other key to continue..." ;
    ch = getch() ;
    if (ch == 27)
      return ;
  }
} while (!valid) ;
if (tcode == 0)
  tcode = bookcodeof(t1code) ;
clrscr() ;
gotoxy(72,1) ;
cout <<"<0>=Exit" ;
BOOK::display(tcode) ;
do
{
  gotoxy(5,13) ;
  clreol() ;
  cout <<"Do you want to modify this record (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
```

```
    if (ch == '0')
      return ;
  } while (ch != 'Y' && ch != 'N') ;
  if (ch == 'N')
    return ;
  gotoxy(5,16) ;
  cout <<"Book Name : " ;
  gotoxy(5,17) ;
  cout <<"Author Name : " ;
  gotoxy(5,18) ;
  cout <<"Publisher : " ;
  gotoxy(5,19);
  cout <<"Address : " ;
  do
  {
    valid = 1 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"Enter the name of the book or <ENTER> for no change" ;
    gotoxy(19,16) ;
    clreol() ;
  gets(tname) ;
  strupr(tname) ;
  if (tname[0] == '0')
    return ;
  if (strlen(tname) > 32)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\7Enter correctly (Range: 1..32)" ;
    getch() ;
  }
} while (!valid) ;
if (strlen(tname) == 0)
  strcpy(tname,bookname(tcode)) ;
do
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
  cout <<"Enter the author's name or <ENTER> for no change" ;
  gotoxy(19,17) ;
  clreol() ;
  gets(tauthor) ;
  strupr(tauthor) ;
  if (tauthor[0] == '0')
    return ;
  if (strlen(tauthor) > 25)
  {
    valid = 0 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"\nEnter correctly (Range: 1..25)" ;
    getch() ;
  }
} while (!valid) ;
if (strlen(tauthor) == 0)
  strcpy(tauthor,authorname(tcode)) ;
do
{
  valid = 1 ;
  gotoxy(5,25) ;
```

```
    clreol() ;
    cout <<"Enter the Publisher's name or <ENTER> for no change" ;
    gotoxy(19,18) ;
    clreol() ;
    gets(tpub) ;
    strupr(tpub) ;
    if (tpub[0] == '0')
      return ;
    if (strlen(tpub) > 33)
    {
      valid = 0 ;
      gotoxy(5,25) ;
      clreol() ;
      cout <<"\nEnter correctly (Range: 1..25)" ;
      getch() ;
    }
  } while (!valid) ;
  if (strlen(tpub) == 0)
  strcpy(tpub,bookpublisher(tcode)) ;
  do
  {
    valid = 1 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"Enter the address of the book or <ENTER> for no change" ;
    gotoxy(19,19) ;
    clreol() ;
    gets(taddr) ;
    strupr(taddr) ;
    if (taddr[0] == '0')
      return ;
    if (strlen(taddr) > 32)
    {
      valid = 0 ;
      gotoxy(5,25) ;
      clreol() ;
      cout <<"\7Enter correctly (Range: 1..32)" ;
      getch() ;
    }
  } while (!valid) ;
  if (strlen(taddr) == 0)
    strcpy(taddr,bookaddr(tcode)) ;
```

```
do
{
  gotoxy(5,25) ;
  clreol() ;
  cout <<" " ;
  gotoxy(5,20) ;
  clreol() ;
  cout <<"Do you want to save changes (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
  if (ch == '0')
    return ;
}while (ch != 'Y' && ch != 'N') ;
if (ch == 'N')
  return ;
BOOK::modify(tcode,tname,tauthor,taddr,tpub) ;
textcolor(BLACK+BLINK) ; textbackground(WHITE) ;
gotoxy(27,22) ;
cprintf(" RECORD MODIFIED ") ;
textcolor(LIGHTGRAY) ;
textbackground(BLACK) ;
```

```
  getch() ;
}


//**************************************************************************
// THIS FUNCTION TO MODIFY THE MEMBER RECORD
//**************************************************************************


void WORKING :: modify_member(void)
{
  MEMBER M ;
  char m1code[10], mname[26], mphone[10], maddress[33], ch ;
  int m2code=0, mcode=0 ;
  int valid ;
  do
  {
    valid = 1 ;
    while (1)
    {
      clrscr() ;
      gotoxy(72,1) ;
  cout <<"<0>=Exit" ;
  gotoxy(5,7) ;
  cout <<"Enter Code no. of the Member to be Modify" ;
  gotoxy(5,8) ;
  cout <<"             or     " ;
  gotoxy(5,9) ;
  cout <<"Press <ENTER> for help " ;
  gets(m1code) ;
  m2code = atoi(m1code) ;
  mcode = m2code ;
  if (m1code[0] == '0')
    return ;
  if (strlen(m1code) == 0)
    M.list() ;
  else
    break ;
}
if (mcode == 0)
{
  valid = 0 ;
  gotoxy(5,25) ;
```

```
    cout <<"\7Enter Correctly" ;
    getch() ;
  }
  if (valid && !member_found(mcode))
  {
    valid = 0 ;
    gotoxy(5,13) ;
    cout <<"\7Record not found" ;
    gotoxy(5,14) ;
    cout <<"Press <ESC> to exit or any other key to continue..." ;
    ch = getch() ;
    if (ch == 27)
      return ;
  }
} while (!valid) ;
clrscr() ;
gotoxy(72,1) ;
cout <<"<0>=Exit" ;
MEMBER::display(mcode) ;
do
{
  gotoxy(5,10) ;
  clreol() ;
  cout <<"Do you want to modify this record (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
  if (ch == '0')
    return ;
}while (ch != 'Y' && ch != 'N') ;
if (ch == 'N')
  return ;
gotoxy(5,13) ;
cout <<"Name : " ;
gotoxy(5,14) ;
cout <<"Phone : " ;
gotoxy(5,15) ;
cout <<"Address : " ;
do
{
  valid = 1 ;
  gotoxy(5,25) ;
  clreol() ;
```

```
    cout <<"Enter the name of the member or <ENTER> for no change" ;

    gotoxy(19,13) ;

    clreol() ;

    gets(mname) ;

    strupr(mname) ;

    if (mname[0] == '0')

      return ;

    if (strlen(mname) > 25)

    {

      valid = 0 ;

      gotoxy(5,25) ;

      clreol() ;

      cout <<"\7Enter correctly (Range: 1..25)" ;

      getch() ;

    }

  }while (!valid) ;

  if (strlen(mname) == 0)

    strcpy(mname,membername(mcode)) ;

  do

  {

    valid = 1 ;
    gotoxy(5,25) ;

    clreol() ;

    cout <<"Enter the Phone no. of Member or <ENTER> for no
change" ;

    gotoxy(19,14) ;

    clreol() ;

    gets(mphone) ;

    if (mphone[0] == '0')

      return ;

    if ((strlen(mphone) < 10 && strlen(mphone) > 0) ||
(strlen(mphone) > 12))

    {

      valid = 0 ;

      gotoxy(5,25) ;

      clreol() ;

      cout <<"\7Enter correctly" ;

      getch() ;

    }

  } while (!valid) ;

  if (strlen(mphone) == 0)

    strcpy(mphone,memberphone(mcode)) ;

  do
  {
    valid = 1 ;
    gotoxy(5,25) ;
    clreol() ;
    cout <<"Enter the address of the member or <ENTER> for no change" ;
    gotoxy(19,15) ;
    clreol() ;
    gets(maddress) ;
    strupr(maddress);
    if(maddress[0]== '0')
      return ;
    if (strlen(maddress) > 32)
    {
      valid = 0 ;
      gotoxy(5,25) ;
      clreol() ;
      cout <<"\7Enter correctly (Range: 1..32)" ;
      getch() ;
    }
  } while (!valid) ;
  if (strlen(maddress) == 0)
```

```
strcpy(maddress,memberaddress(mcode)) ;
gotoxy(5,25) ;
clreol() ;
do
{
  gotoxy(5,18) ;
  clreol() ;
  cout <<"Do you want to save changes (y/n) : " ;
  ch = getche() ;
  ch = toupper(ch) ;
  if (ch == '0')
    return ;
} while (ch != 'Y' && ch != 'N') ;
if (ch == 'N')
  return ;
MEMBER::modify(mcode,mname,mphone,maddress) ;
textcolor(BLACK+BLINK) ;
textbackground(WHITE) ;
gotoxy(27,20) ;
cprintf(" RECORD MODIFIED ") ;
textcolor(LIGHTGRAY) ;
```

```
  textbackground(BLACK) ;
  getch() ;
}


//********************************************************
*********************
// THIS FUNCTION TO DELETE THE BOOK RECORD
//********************************************************
*********************


void WORKING :: delete_book(void)
{
  BOOK B ;
  char t1code[5], tname[33], tauthor[26], ch ;
  int t2code=0, tcode=0 ;
  int valid ;
  do
  {
    valid = 1 ;
    while (1)
    {
      clrscr() ;
      gotoxy(72,1) ;
      cout <<"<0>=Exit" ;
      gotoxy(5,5) ;
      cout <<"Enter Code or Name of the Book to be Deleted" ;
      gotoxy(5,6) ;
      cout <<"    or    " ;
      gotoxy(5,7) ;
      cout <<"Press <ENTER> for help " ;
      gets(t1code) ;
      if (t1code[0] == '0')
        return ;
      if (strlen(t1code) == 0)
        B.list() ;
      else
        break ;
    }
    t2code = atoi(t1code) ;
    tcode = t2code ;
    if ((tcode == 0 && !bookname_found(t1code)) || (tcode != 0 && !book_found(tcode)))
```

```
  {
    valid = 0 ;
    gotoxy(5,10) ;
    cout <<"\7Record not found" ;
    gotoxy(5,11) ;
    cout <<"Press <ESC> to exit or any other key to continue..." ;
    ch = getch() ;
    if (ch == 27)
      return ;
  }
} while (!valid) ;
if (tcode == 0)
  tcode = bookcodeof(t1code) ;
clrscr() ;
gotoxy(72,1) ;
cout <<"<0>=Exit" ;
BOOK::display(tcode) ;
do
{
  gotoxy(5,13) ;
  clreol() ;
```

```
    cout <<"Do you want to delete this record (y/n) : " ;
    ch = getche() ;
    ch = toupper(ch) ;
    if (ch == '0')
      return ;
  }while (ch != 'Y' && ch != 'N') ;
  if (ch == 'N')
    return ;
  BOOK::delete_rec(tcode) ;
  textcolor(BLACK+BLINK) ;
  textbackground(WHITE) ;
  gotoxy(27,22) ;
  cprintf(" RECORD DELETED ") ;
  textcolor(LIGHTGRAY) ;
  textbackground(BLACK) ;
  getch() ;
}


//************************************************************
*********************

// THIS FUNCTION TO DELETE THE MEMBER RECORD.
```

```
//********************************************************
*********************


void WORKING :: delete_member(void)
{
  MEMBER M ;
  char m1code[5], mname[26], mphone[10], maddress[33], ch ;
  int m2code=0, mcode=0 ;
  int valid ;
  do
  {
    valid = 1 ;
    while (1)
    {
      clrscr() ;
      gotoxy(72,1) ;
      cout <<"<0>=Exit" ;
      gotoxy(5,7) ;
      cout <<"Enter Code no. of the Member to be Deleted" ;
      gotoxy(5,8) ;
      cout <<"    or    " ;
      gotoxy(5,9) ;
      cout <<"Press <ENTER> for help " ;
      gets(m1code) ;
      m2code = atoi(m1code) ;
      mcode = m2code ;
      if (m1code[0] == '0')
        return ;
      if (strlen(m1code) == 0)
        M.list() ;
      else
        break ;
    }
    if (mcode == 0)
    {
      valid = 0 ;
      gotoxy(5,25) ;
      cout <<"\7Enter Correctly" ;
      getch() ;
    }
    if (valid && !member_found(mcode))
    {
      valid = 0 ;
      gotoxy(5,13) ;
      cout <<"\7Record not found" ;
      gotoxy(5,14) ;
      cout <<"Press <ESC> to exit or any other key to continue..." ;
      ch = getch() ;
      if (ch == 27)
        return ;
    }
   } while (!valid) ;
   clrscr() ;
   gotoxy(72,1) ;
   cout <<"<0>=Exit" ;
   MEMBER::display(mcode) ;
   do
   {
    gotoxy(5,10) ;
    clreol() ;
    cout <<"Do you want to Delete this record (y/n) : " ;
    ch = getche() ;
    ch = toupper(ch) ;
    if (ch == '0')
```

```
      return ;
  } while (ch != 'Y' && ch != 'N') ;
  if (ch == 'N')
    return ;
  MEMBER::delete_rec(mcode) ;
  textcolor(BLACK+BLINK) ;
  textbackground(WHITE) ;
  gotoxy(27,22) ;
  cprintf(" RECORD DELETED ") ;
  textcolor(LIGHTGRAY) ;
  textbackground(BLACK) ;
  getch() ;
}


//**************************************************************
*********************
// MAIN FUNCTION CALLING INTRODUCTION AND MAIN MENU.
//**************************************************************
*********************
```

```
void main(void)
{
  checkdate(); //TO CHECK THE DATE TO DELETE BOOK
  MENU menu ;
  menu.intro() ;
  menu.main_menue() ;
}
```

USER DEFINED HEADER FILES

```
//****************************************************************************

//****************************************************************************

//HEADER FILE: DATE.H

//****************************************************************************

//****************************************************************************


#include <fstream.h>

#include<conio.h>

#include<stdio.h>

#include<dos.h>

#include<link.h> //USER DEFINED HEADER FILE


int d2, m2, y2;

void deldate (int day2, int month2, int year2);


//****************************************************************************

//FUNCTION TO GET SYSTEM DATE AND SEND IT TO ANOTHER FUNCTION.
```

```
//*******************************************************
********************

void storedate ()

{

 struct date d;

 date date;

 getdate (&d);

 int day2, month2, year2;

 struct date dat;

 getdate (&dat); day2 = dat.da_day;

 month2 = dat.da_mon;

 year2 = dat.da_year;

 day2 = day2 + 15;

 if (((month2 == 1) || (month2 == 3) || (month2 == 5) ||
(month2 == 7)|| (month2 == 8) || (month2 == 10) || (month2
== 12)) && ((day2 > 31)))

  {

   day2 = day2 - 31;

   month2 = month2 + 1;

   if (month2 == 13)

    {

     year2++;
     month2 = 1;
    }
  }
 else if (((month2 == 4) || (month2 == 6) || (month2 == 9) ||
(month2 == 11)) && ((day2 > 30)))
  {
   day2 -= 30;
   month2++;
  }
 else if (((year2 % 4 == 0) || (year2 % 400 == 0)) && (day2 >
29) && (month2 == 2))
  {
   day2 -= 29;
   month2++;
  }
 else if (((year2 % 4 != 0) || (year2 % 100 != 0) || (year2 % 400
!= 0)) && (day2 > 28) && (month2 == 2))
  {
   day2 -= 28;
   month2++;
  }
 cout << endl;
 deldate (day2, month2, year2);
}


//***********************************************************
********************
//FUNCTON TO STORE DELETION DATE OF BOOK.
//***********************************************************
********************
void deldate (int day2, int month2, int year2)
{
 fstream f;
 date dat;
 f.open ("download.txt", ios::in | ios::out);
 system("attrib +H download.txt");
 d2 = day2;
 m2 = month2;
 y2 = year2;
 f.write ((char *) &dat, sizeof (dat));
 f.close ();
}
```

```
//********************************************************
********************

//FUNCTION TO DELETE THE LINK OF BOOK IF THE TIME IS UP
OF USER.

//********************************************************
********************

void checkdate ()

{

 fstream f;

 date d;

 date date;

 getdate (&d);

 int d1, m1, y1;

 d1 = d.da_day;

 m1 = d.da_mon;

 y1 = d.da_year;

 f.open ("download.txt", ios::in | ios::out);

 !f.eof ();

 f.read ((char *) &d, sizeof (d));

 if (d1 >= d2 && m1 >= m2 && y1 >= y2)

  {

   link ("nobook.txt", "no_file");
   f.close ();
  }
 else
  return;
}
```

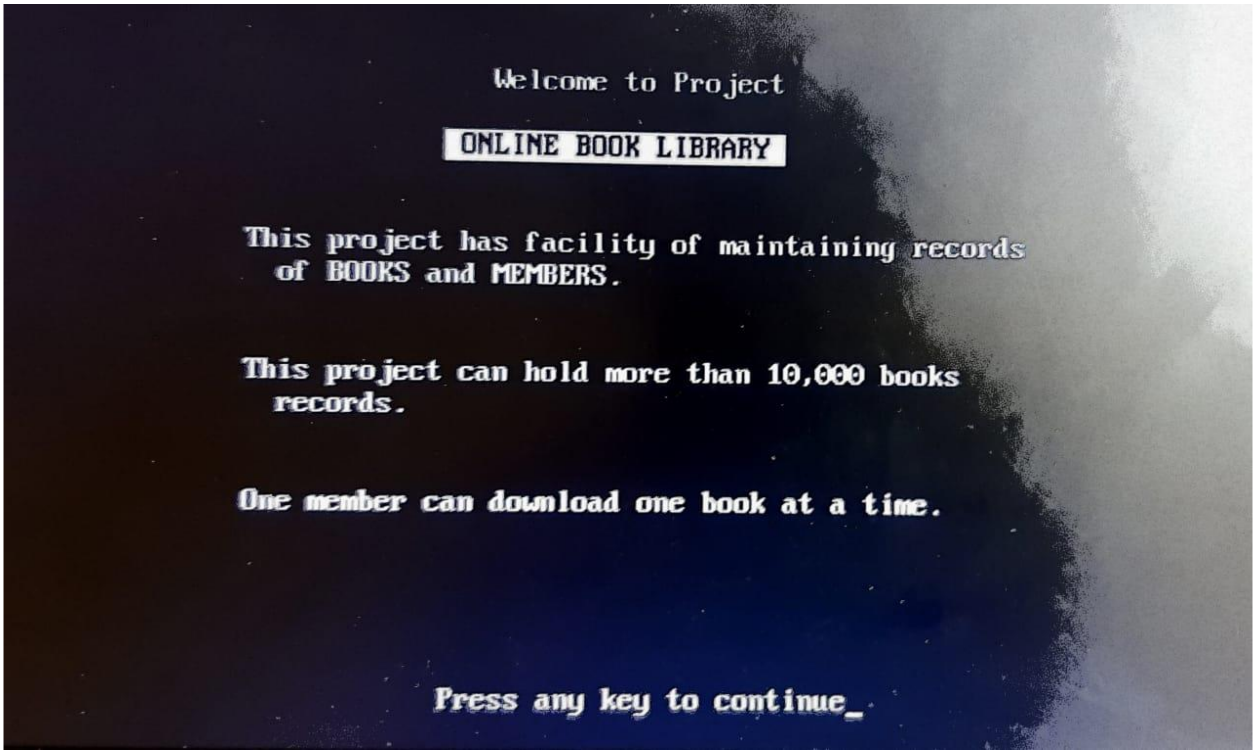
```
//****************************************************************************

//****************************************************************************

//HEADER FILE: COLBOX.H

//****************************************************************************

//****************************************************************************


#include<iostream.h>

#include <conio.h>

#include <dos.h>



//****************************************************************************

//FUNCTION TO MAKE COLOR BOX.

//****************************************************************************

void colorbox(int R1,int C1,int R2,int C2)

{

 for(int i=R1;i<=R2;i++)
```

```
  {
   textcolor(i);
   gotoxy(i,C1);
   cprintf("%c",219); //to print VERTICAL LINE
   gotoxy(i,C2);
   cprintf("%c",219);
  }
 for(int j=C1;j<=C2;j++)
  {
   textcolor(j);
   gotoxy(R1,j);
   cprintf("%c",219); //to print HORIZONTAL LINE
   gotoxy(R2,j);
   cprintf("%c",219);
  }
 textcolor(LIGHTGRAY) ;
}
```

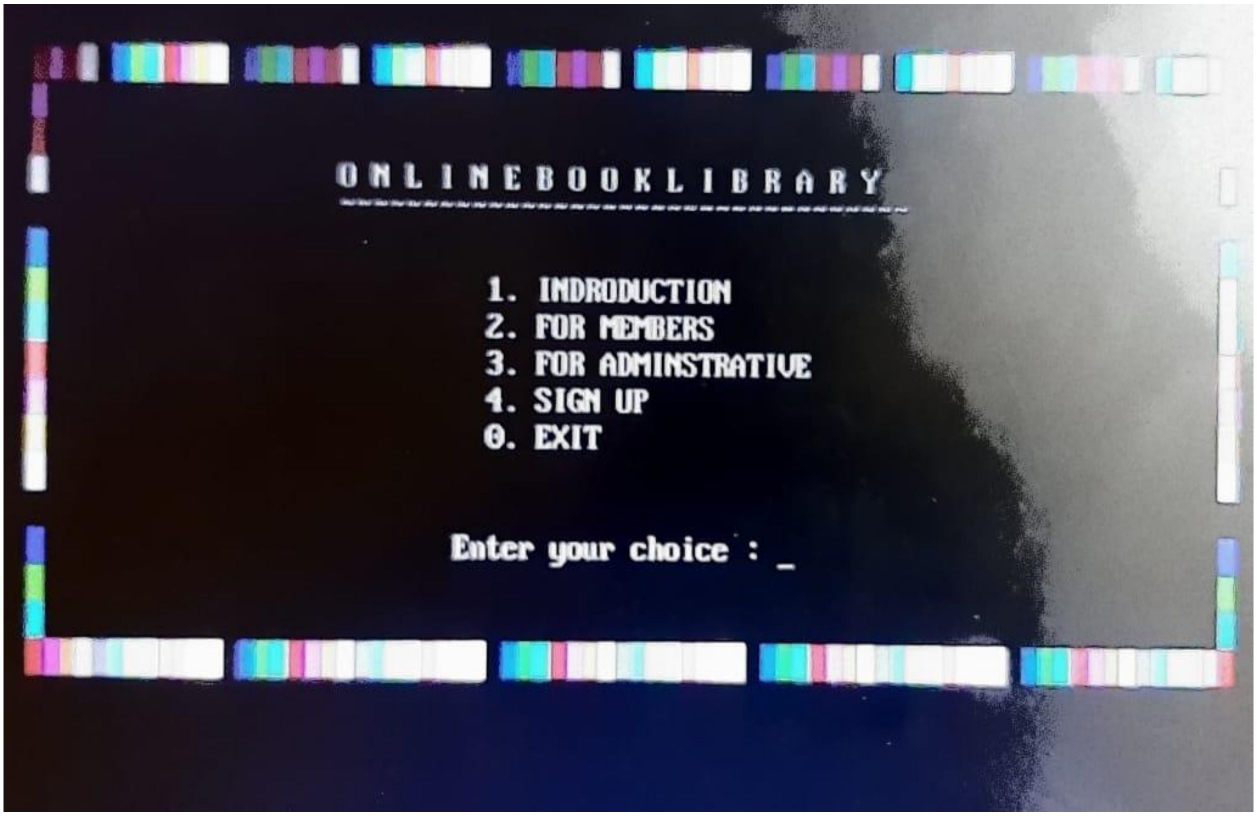
```
//********************************************************
********************

//********************************************************
********************

//HEADER FILE: LINK.H

//********************************************************
********************

//********************************************************
********************


#include<iostream.h>

#include<stdlib.h>

#include<string.h>

#include<stdio.h>

#include<conio.h>

#include<process.h>



//********************************************************
********************

//FUNCTION TO MAKE THE LINK OF BOOK.

//********************************************************
********************
```

```
void link (char*bookurl,char*bookname)
{
FILE *stream;
stream = fopen ("BOOK.HTML", "w");
if (stream != NULL)
 {
  fprintf (stream, "<HTML>\n");
  fprintf (stream, "<ALIGN=CENTER>BOOK LINK IS BELOW:<H2>\n");
  fprintf (stream, "<ALIGN=CENTER>\n");
  fprintf (stream, "<A HREF=\"%s\">%s</A>\n", bookurl,bookname);
  fprintf (stream, "<H2>\n");
  fprintf (stream, "</FONT>\n");
  fprintf (stream, "</BODY>\n");
  fprintf (stream, "</HTML>");
 }
}
```

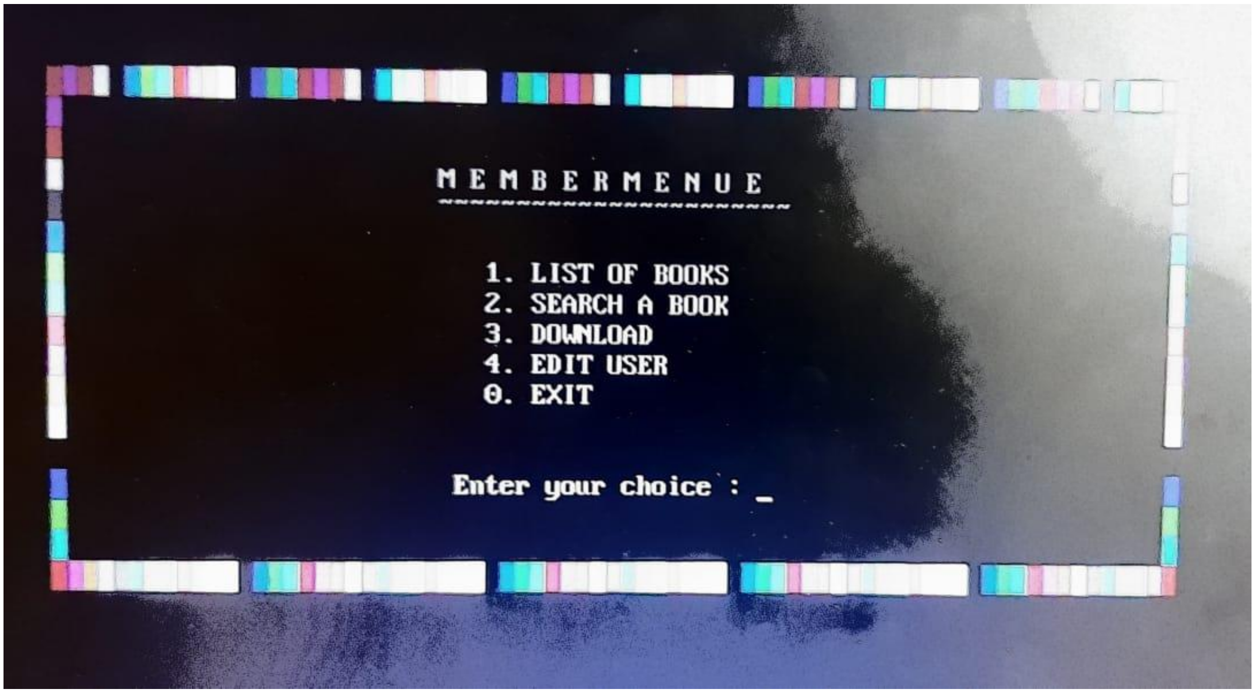
OUTPUT



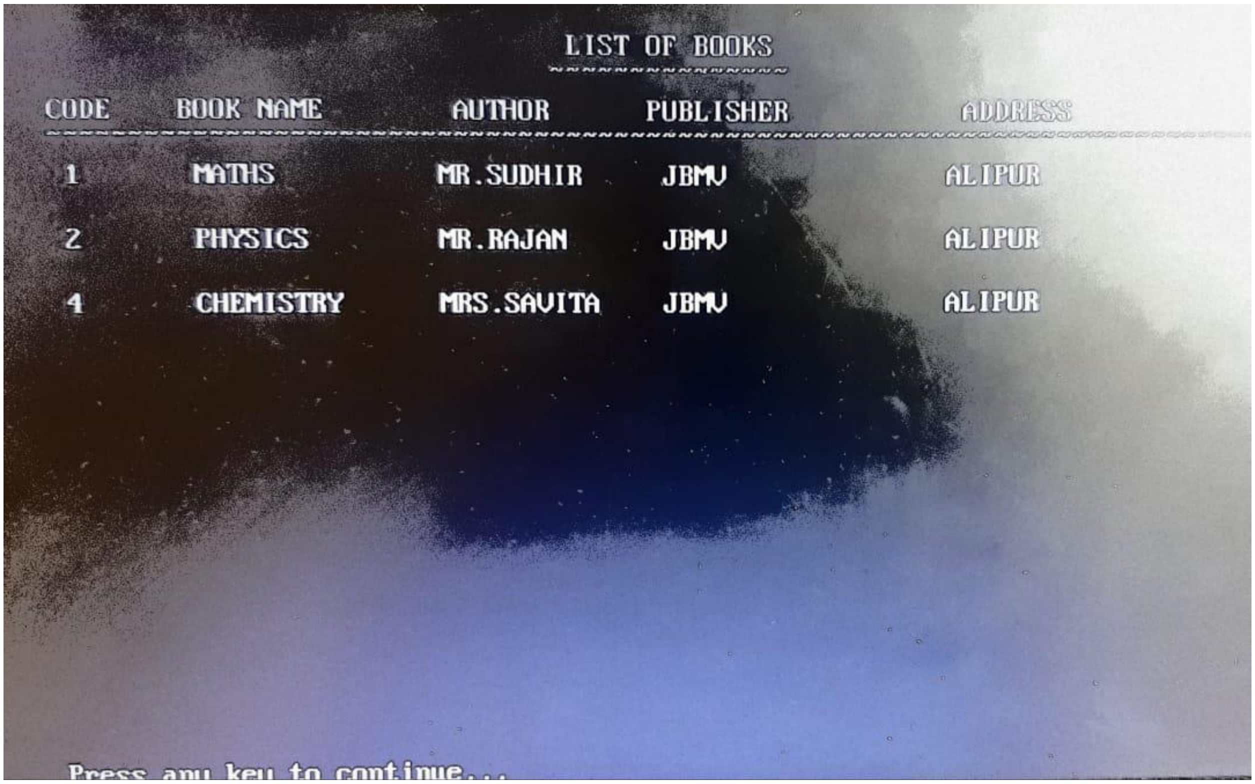
```
                    Welcome to Project

                   ONLINE BOOK LIBRARY

      This project has facility of maintaining records
        of BOOKS and MEMBERS.

      This project can hold more than 10,000 books
        records.

      One member can download one book at a time.

                 Press any key to continue_
```



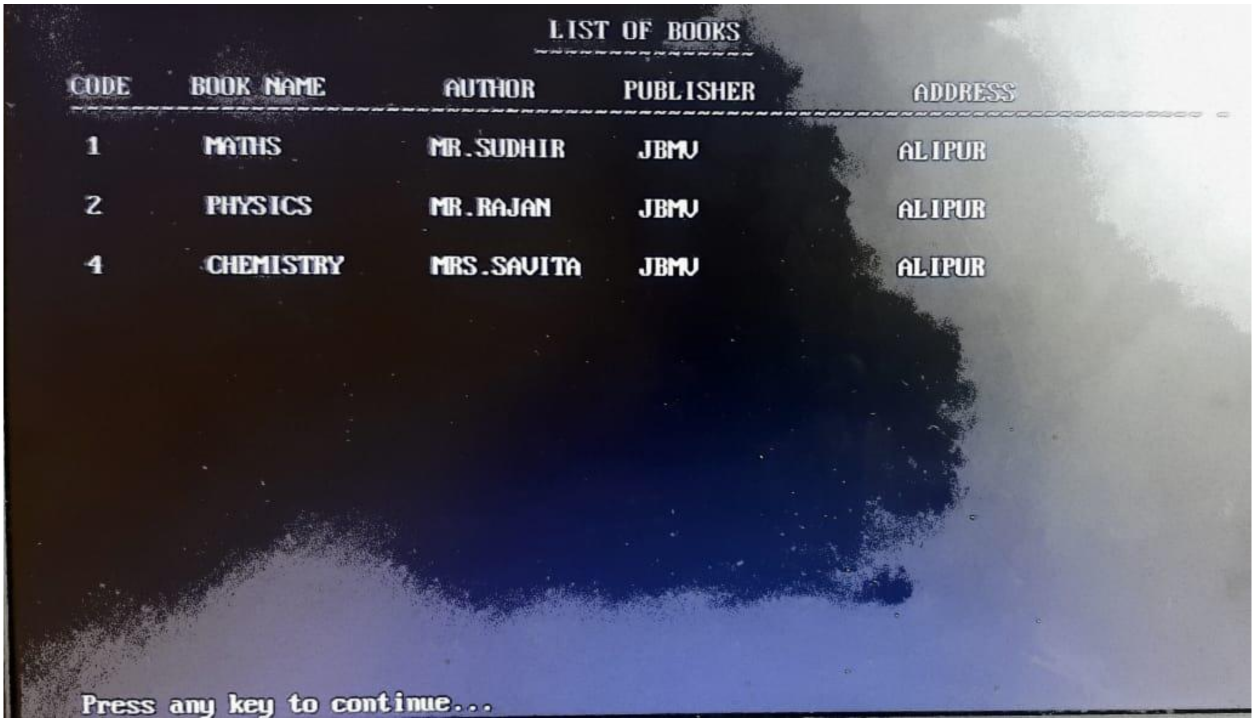
```
            O N L I N E B O O K L I B R A R Y
            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                   1. INDRODUCTION
                   2. FOR MEMBERS
                   3. FOR ADMINSTRATIVE
                   4. SIGN UP
                   0. EXIT

               Enter your choice : _
```



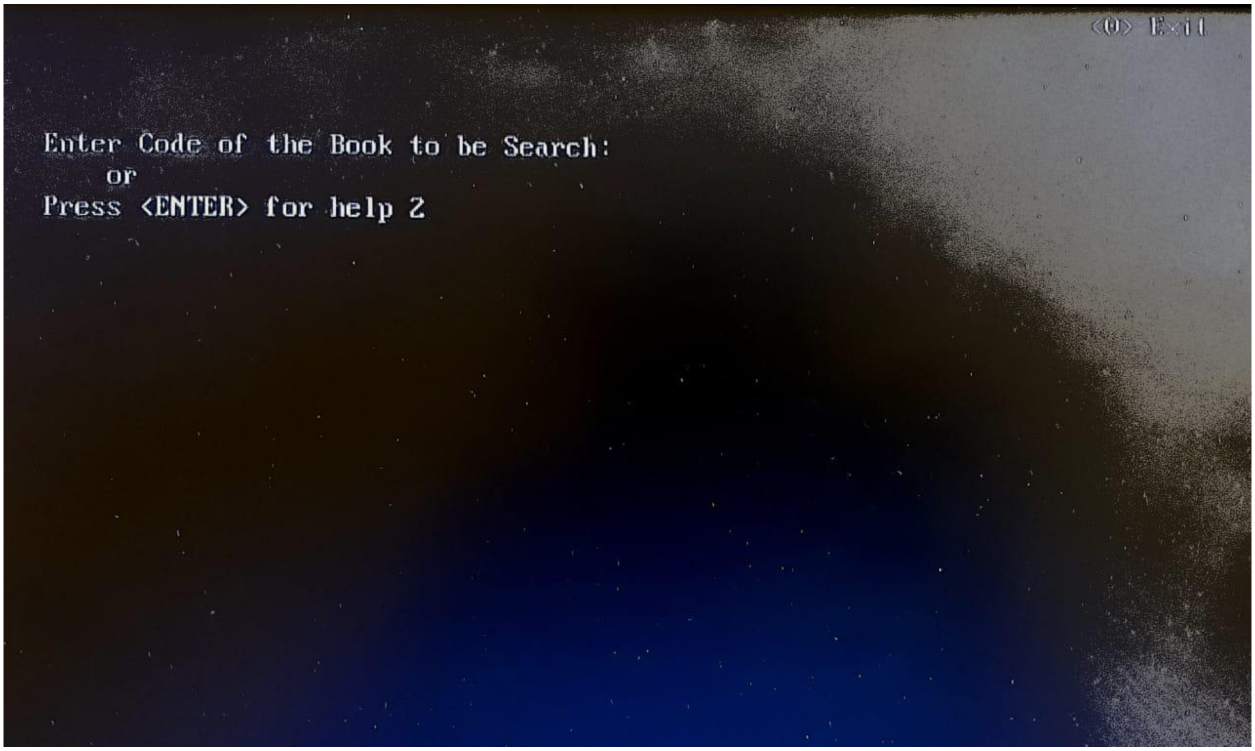
```
                    M E M B E R M E N U
                    ~~~~~~~~~~~~~~~~~~~~~~~

                      1. LIST OF BOOKS
                      2. SEARCH A BOOK
                      3. DOWNLOAD
                      4. EDIT USER
                      0. EXIT

                   Enter your choice : _
```



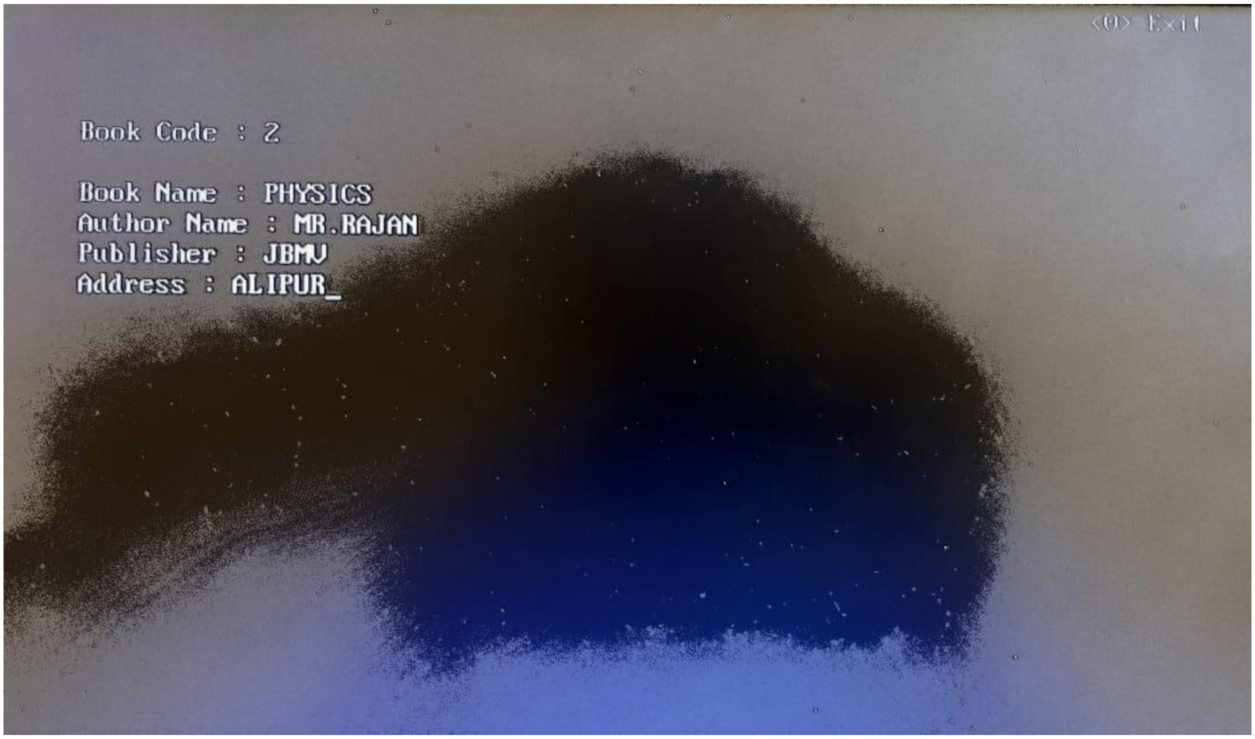
```
                              LIST OF BOOKS
                              ~~~~~~~~~~~~~~~~
CODE    BOOK NAME         AUTHOR        PUBLISHER          ADDRESS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 1       MATHS           MR.SUDHIR       JBMV             ALIPUR

 2       PHYSICS         MR.RAJAN        JBMV             ALIPUR

 4       CHEMISTRY       MRS.SAVITA      JBMV             ALIPUR



 Press any key to continue...
```



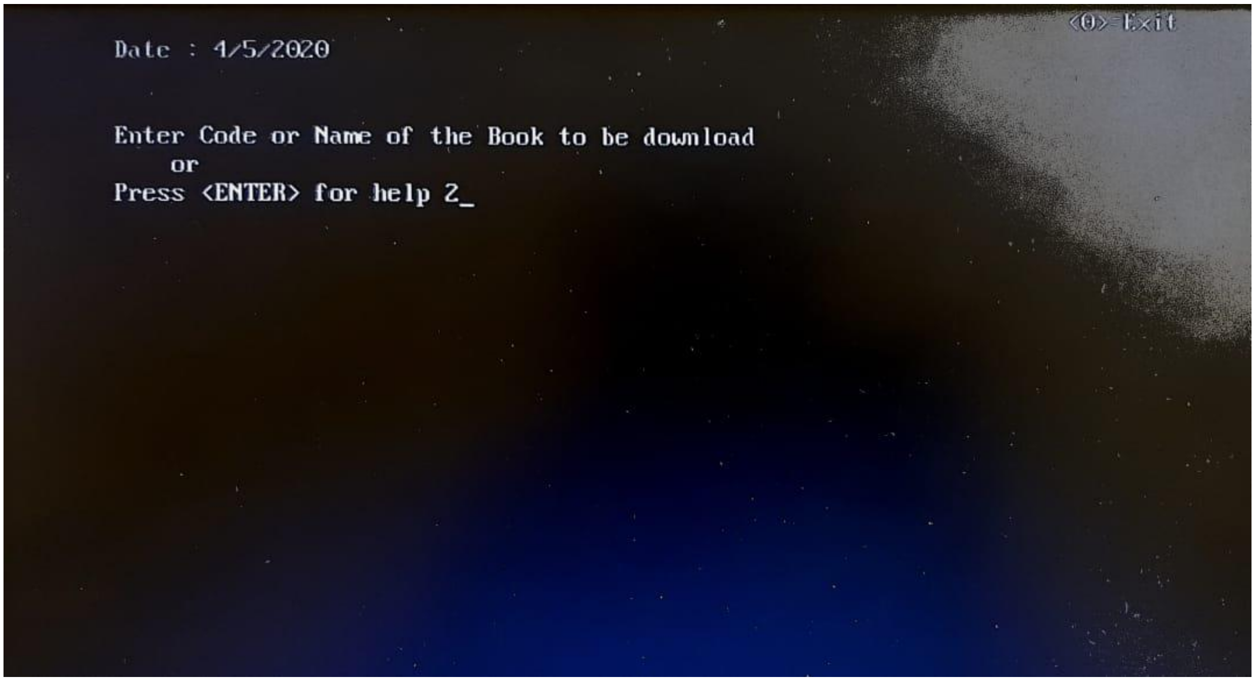
```
                         LIST OF BOOKS
                         ~~~~~~~~~~~~~~~
  CODE   BOOK NAME       AUTHOR       PUBLISHER          ADDRESS
  ------------------------------------------------------------------------
   1      MATHS         MR.SUDHIR      JBMV             ALIPUR

   2      PHYSICS       MR.RAJAN       JBMV             ALIPUR

   4      CHEMISTRY     MRS.SAVITA     JBMV             ALIPUR





   Press any key to continue...
```



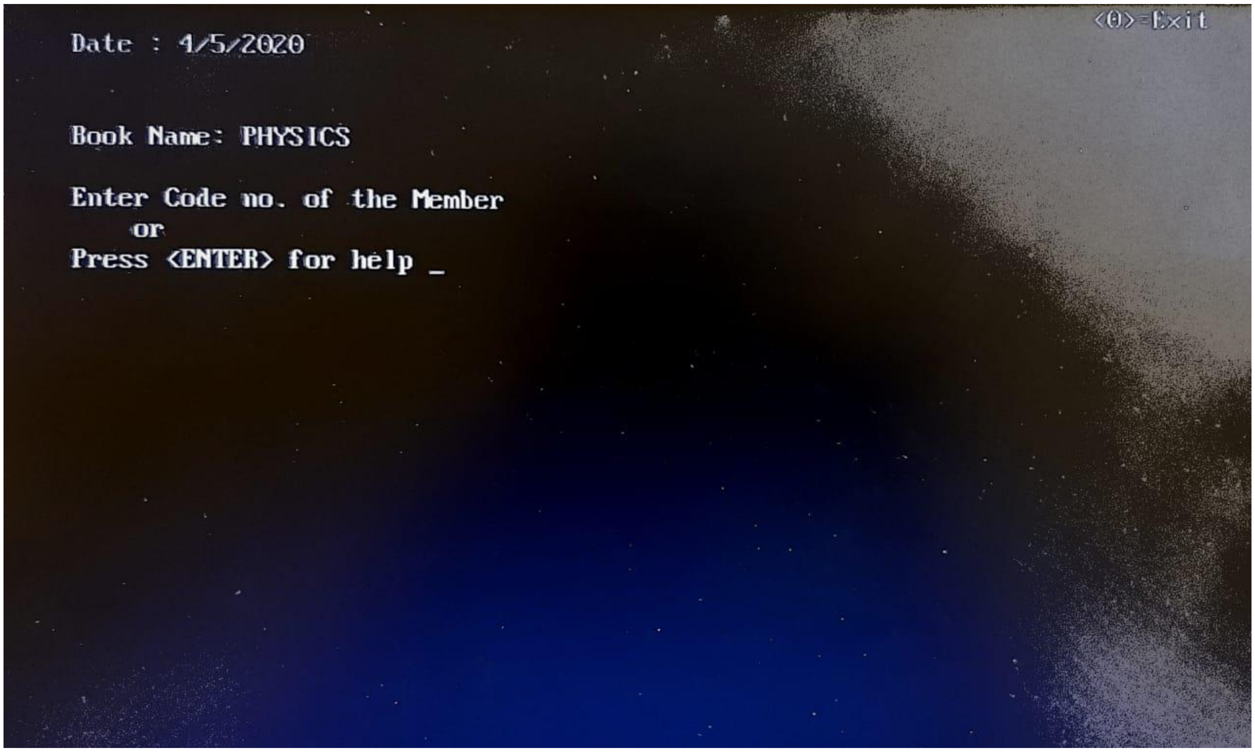
```
                                                            <0> Exit


Enter Code of the Book to be Search:
    or
Press <ENTER> for help 2
```



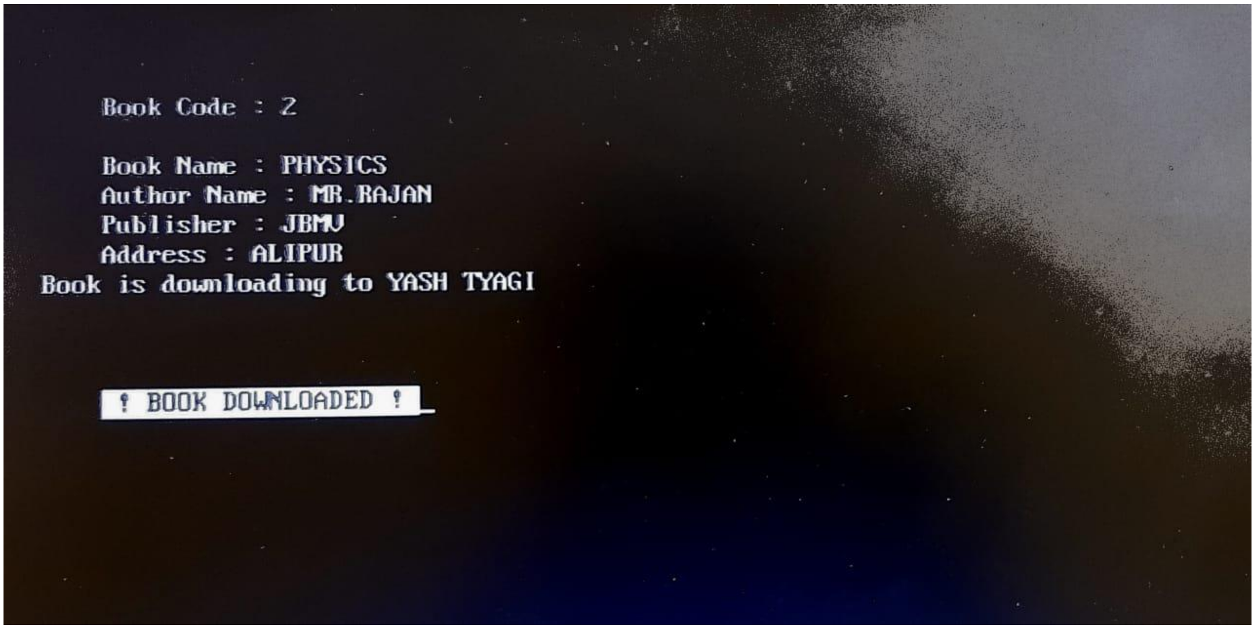
```
                                                              <0> Exit



    Book Code : 2

    Book Name : PHYSICS
    Author Name : MR.RAJAN
    Publisher : JBMV
    Address : ALIPUR_
```



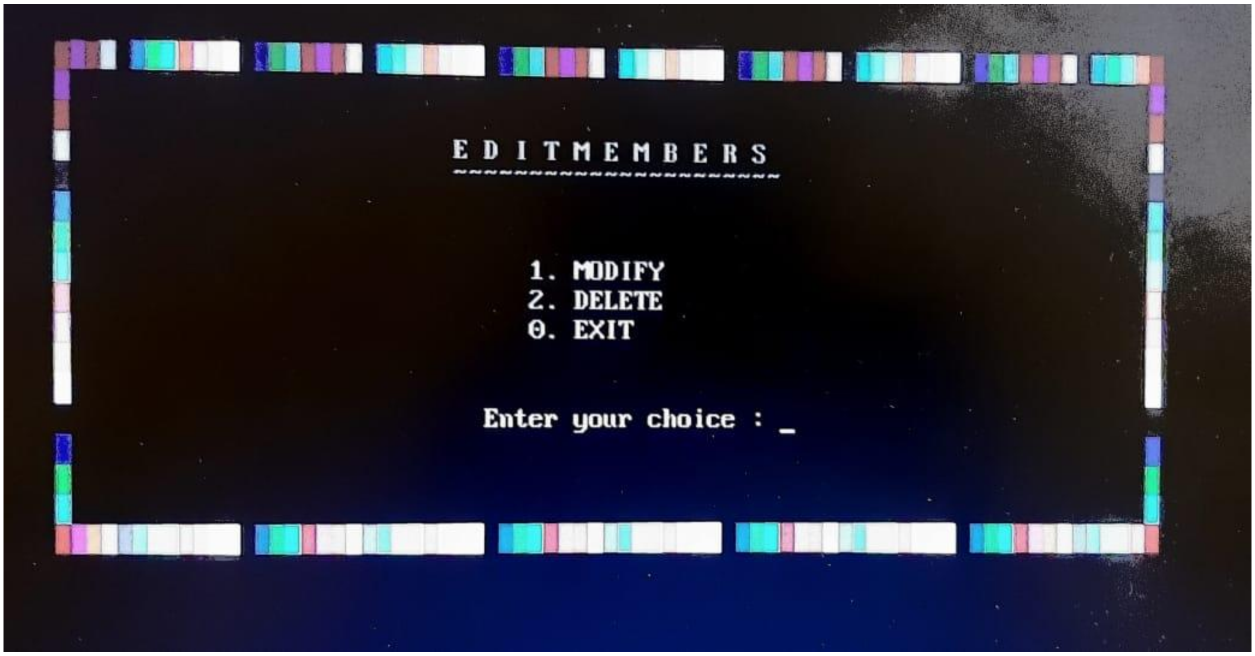
```
                                                              <0> Exit
      Date : 4/5/2020


      Enter Code or Name of the Book to be download
          or
      Press <ENTER> for help 2_
```



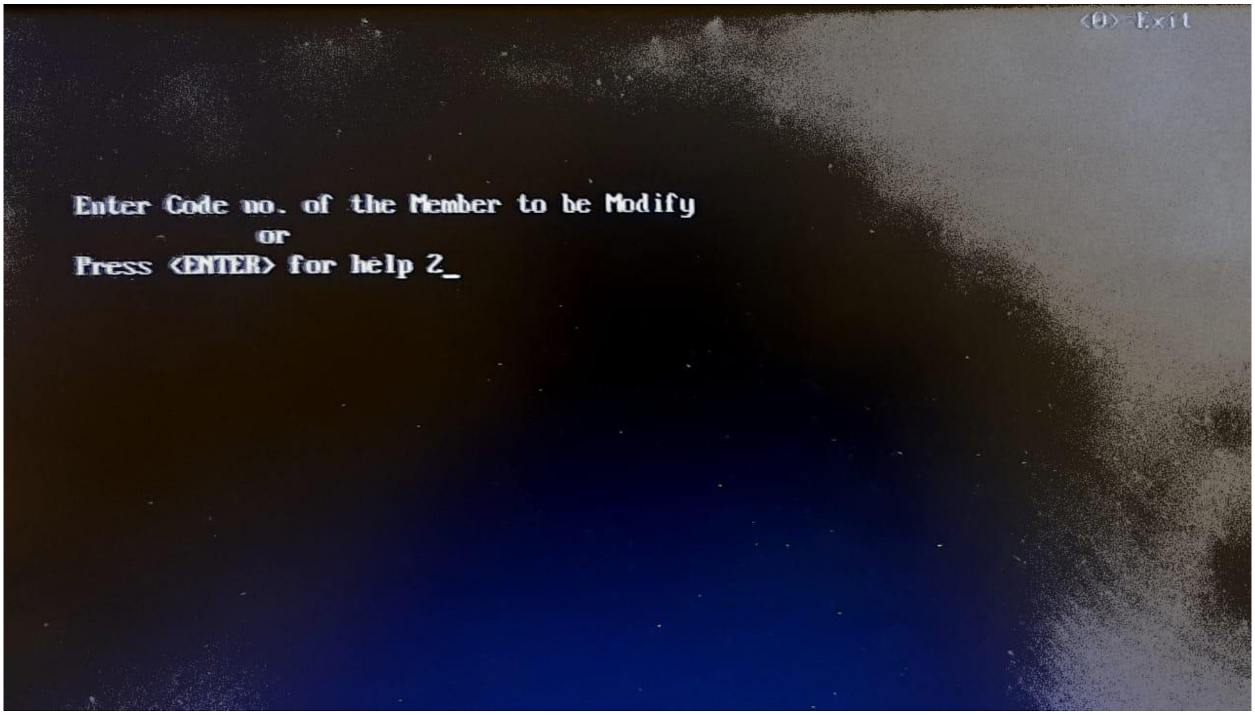
```
                                                    <0>=Exit
  Date : 4/5/2020

  Book Name: PHYSICS

  Enter Code no. of the Member
      or
  Press <ENTER> for help _
```



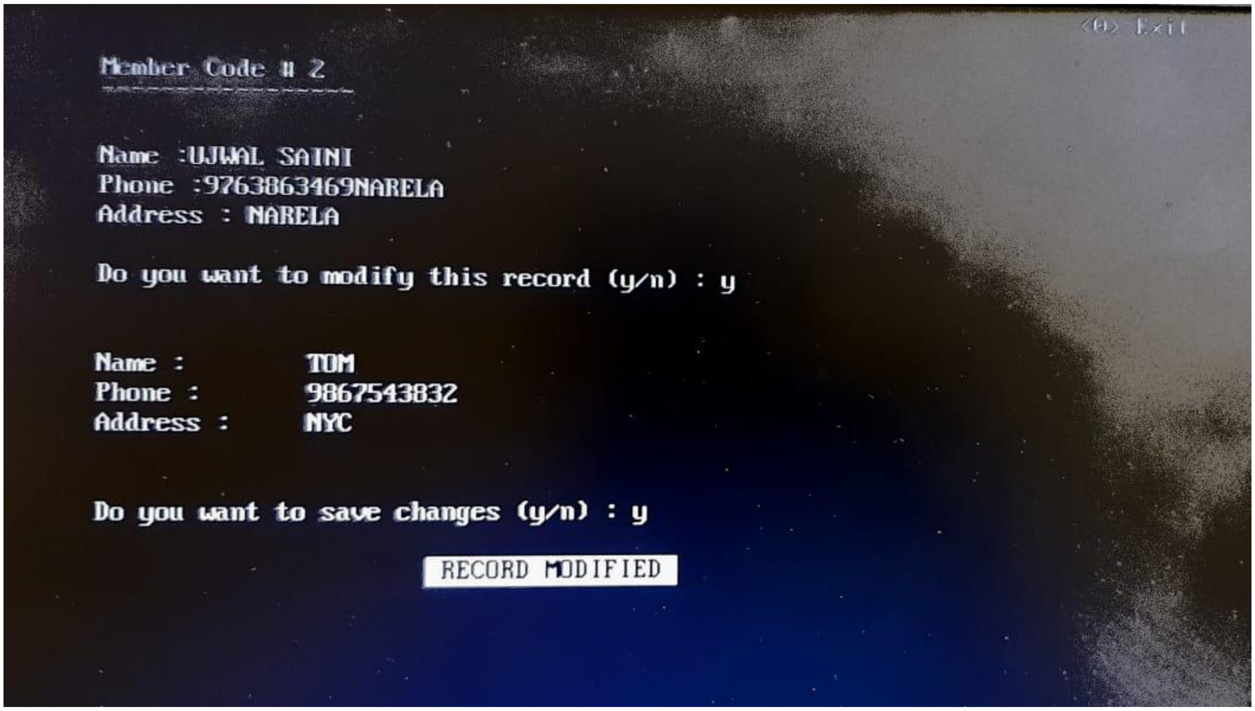
```
    Book Code : 2

    Book Name : PHYSICS
    Author Name : MR.RAJAN
    Publisher : JBMV
    Address : ALIPUR
Book is downloading to YASH TYAGI


    ! BOOK DOWNLOADED ! _
```



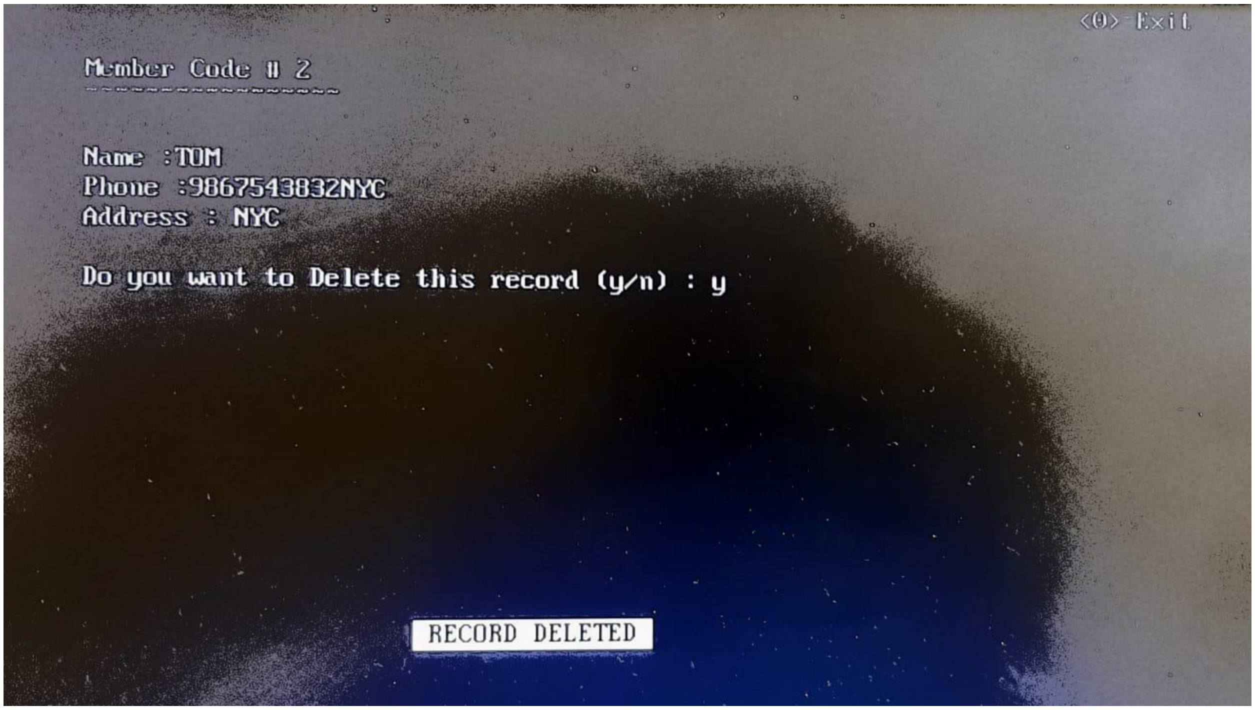
```
                    EDITMEMBERS
                    ~~~~~~~~~~~~~~~~~~~~~~

                        1. MODIFY
                        2. DELETE
                        0. EXIT

                    Enter your choice : _
```



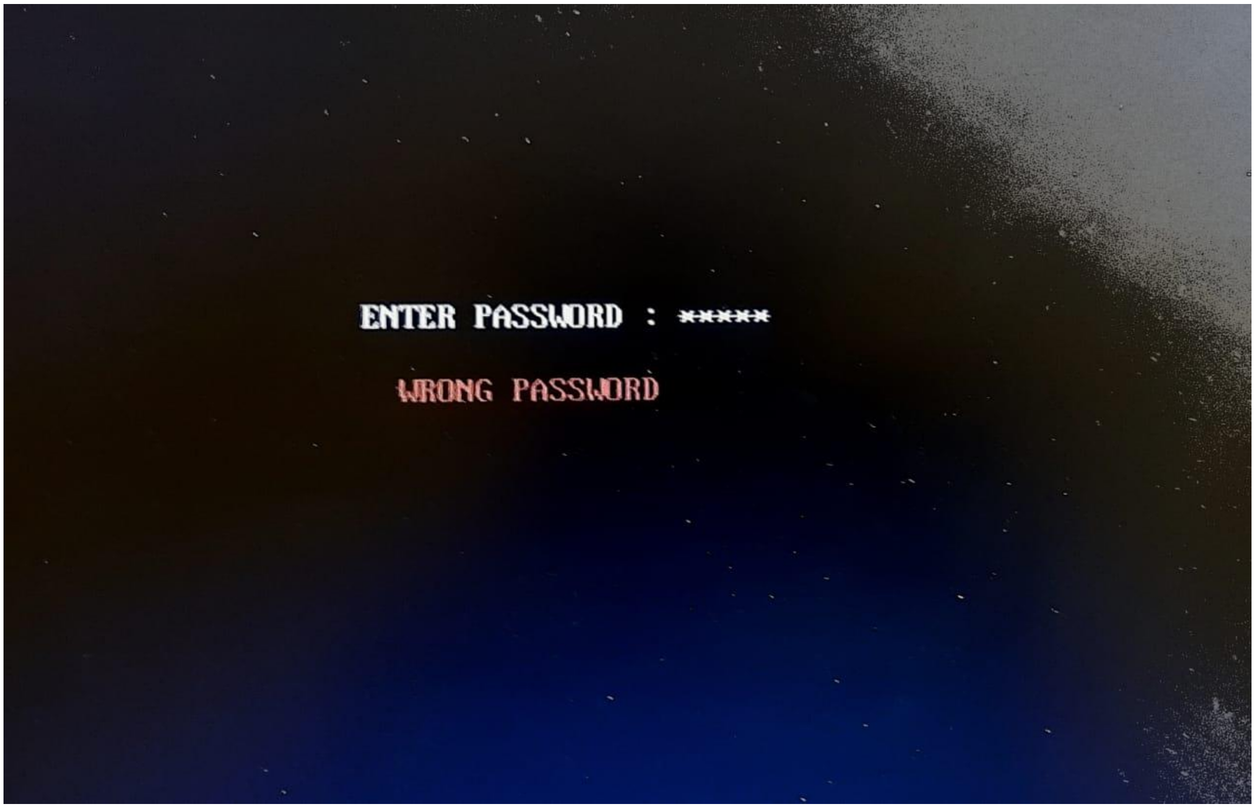
```
                                                          <0>-Exit



Enter Code no. of the Member to be Modify
           or
Press <ENTER> for help 2_
```



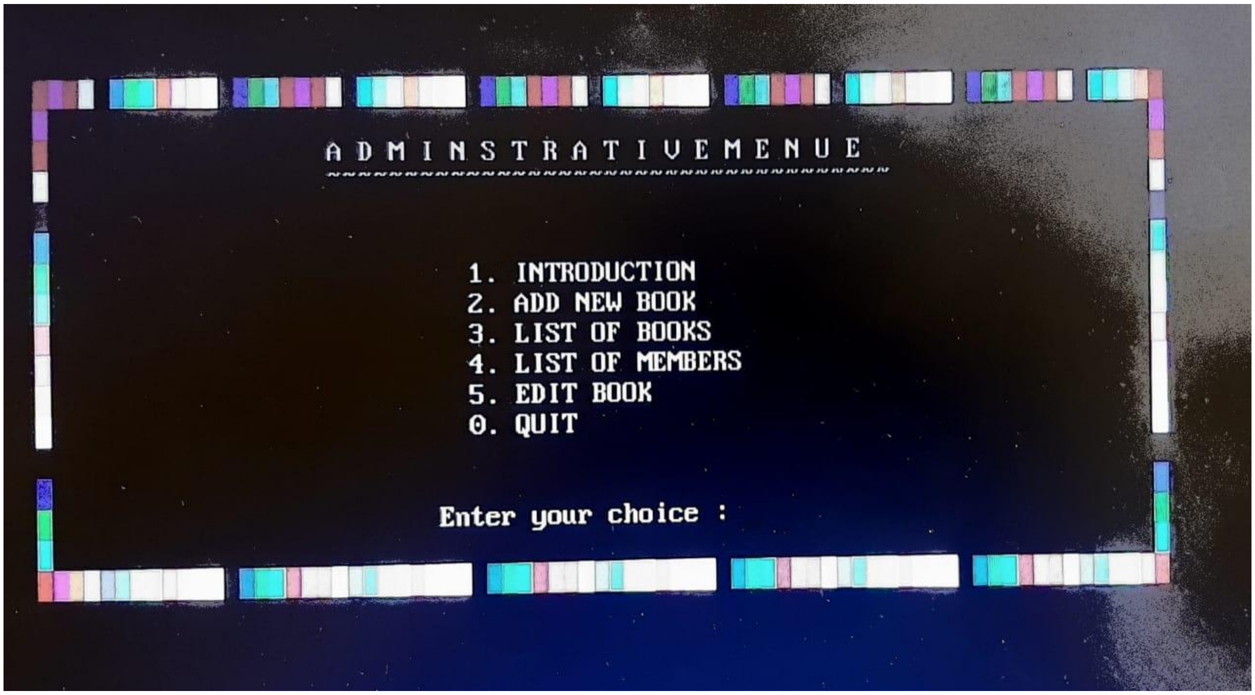
```
                                                              <0> - Exit
   Member Code # 2
   ~~~~~~~~~~~~~~~~~~

   Name  :UJWAL SAINI
   Phone  :9763863469NARELA
   Address  : NARELA

   Do you want to modify this record (y/n) : y


   Name :          TOM
   Phone :         9867543832
   Address :       NYC


   Do you want to save changes (y/n) : y
                          RECORD MODIFIED
```



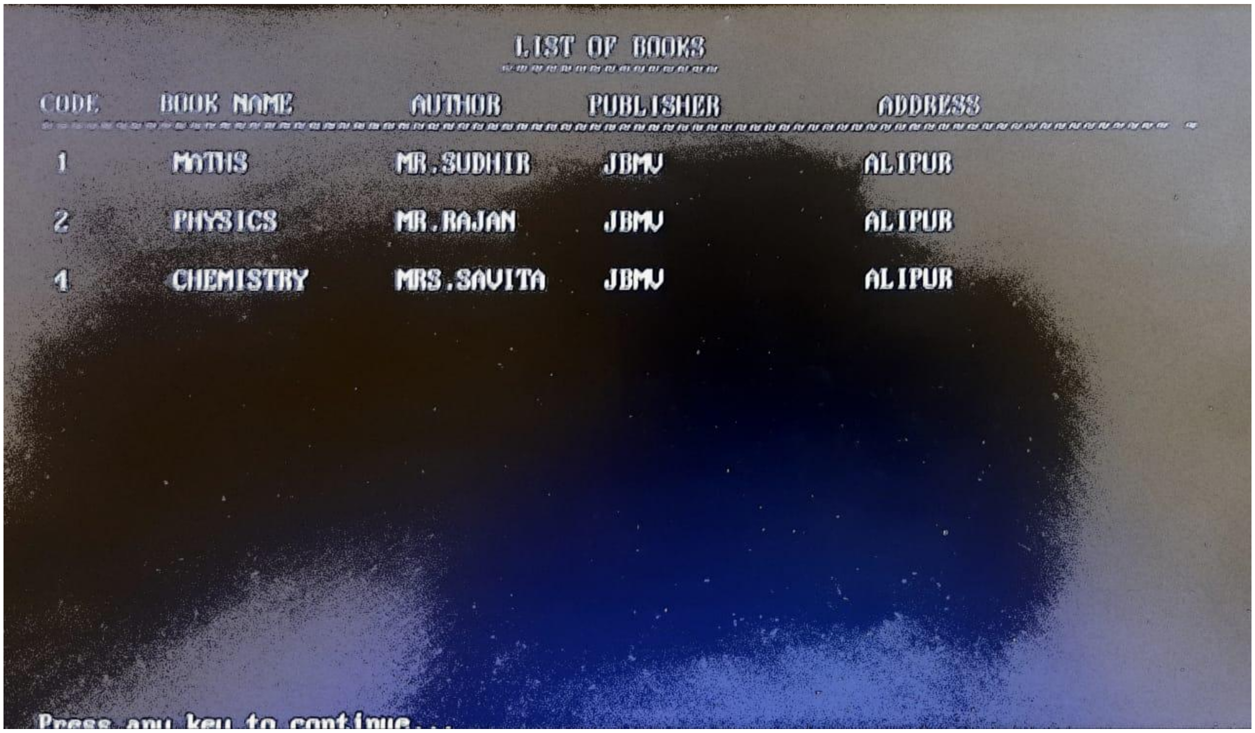
```
                                                              <0> - Exit
  Member Code # 2
  ~~~~~~~~~~~~~~~~~~

  Name  :TOM
  Phone  :9867543832NYC
  Address  : NYC

  Do you want to Delete this record (y/n) : y







                         RECORD DELETED
```



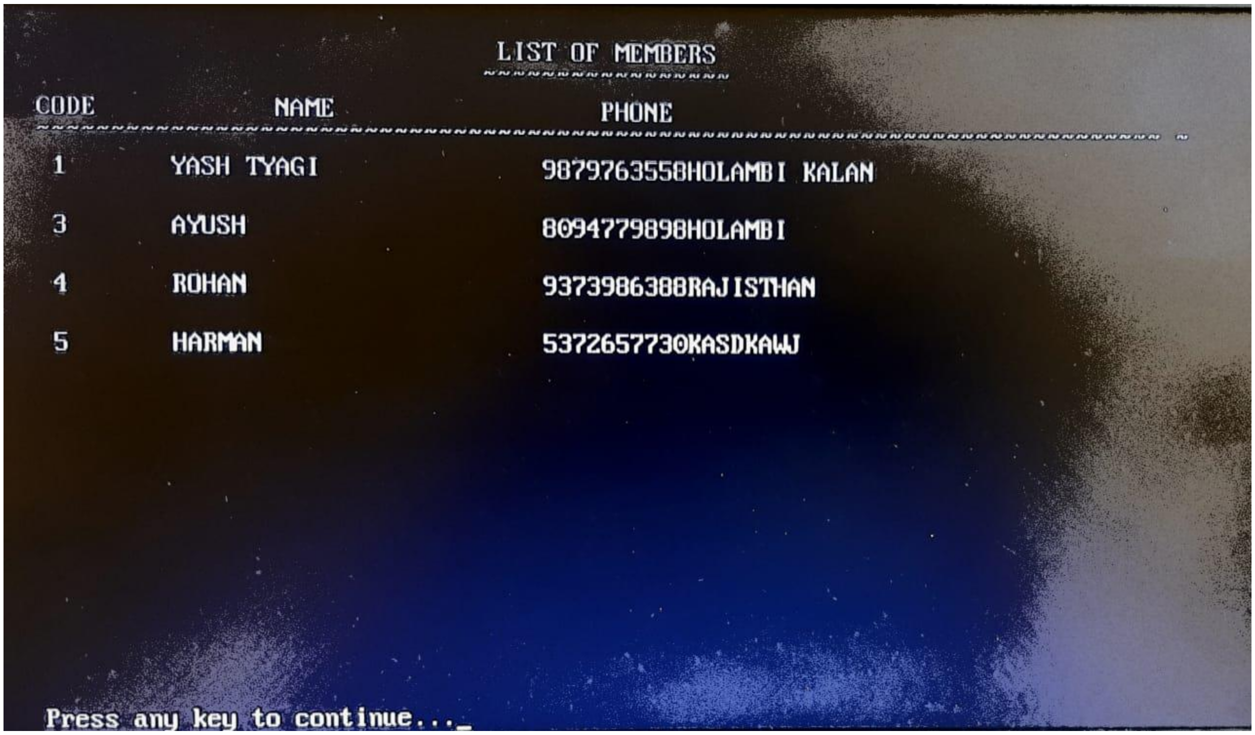
```
ENTER PASSWORD : *****

  WRONG PASSWORD
```



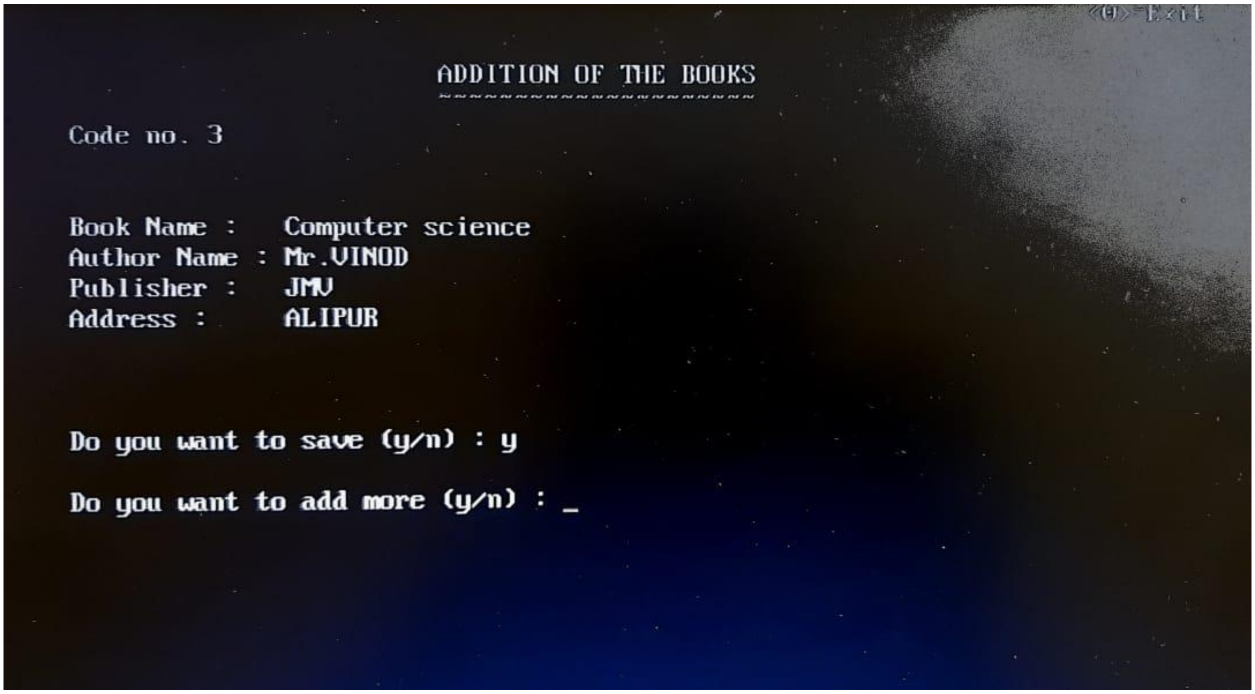
```
            ADMINSTRATIVEMENUE

               1. INTRODUCTION
               2. ADD NEW BOOK
               3. LIST OF BOOKS
               4. LIST OF MEMBERS
               5. EDIT BOOK
               0. QUIT

             Enter your choice :
```



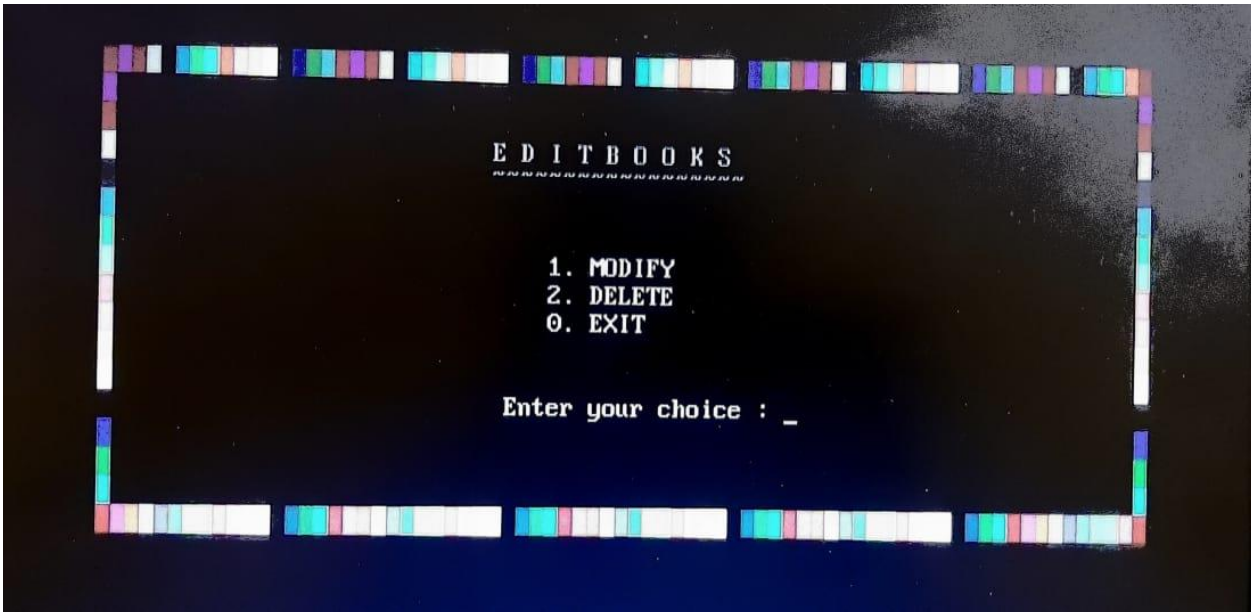
```
                              LIST OF BOOKS
                             ~~~~~~~~~~~~~~~~~
CODE    BOOK NAME        AUTHOR        PUBLISHER           ADDRESS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 1       MATHS          MR.SUDHIR      JBMV               ALIPUR

 2       PHYSICS        MR.RAJAN       JBMV               ALIPUR

 4       CHEMISTRY      MRS.SAVITA     JBMV               ALIPUR






Press any key to continue...
```



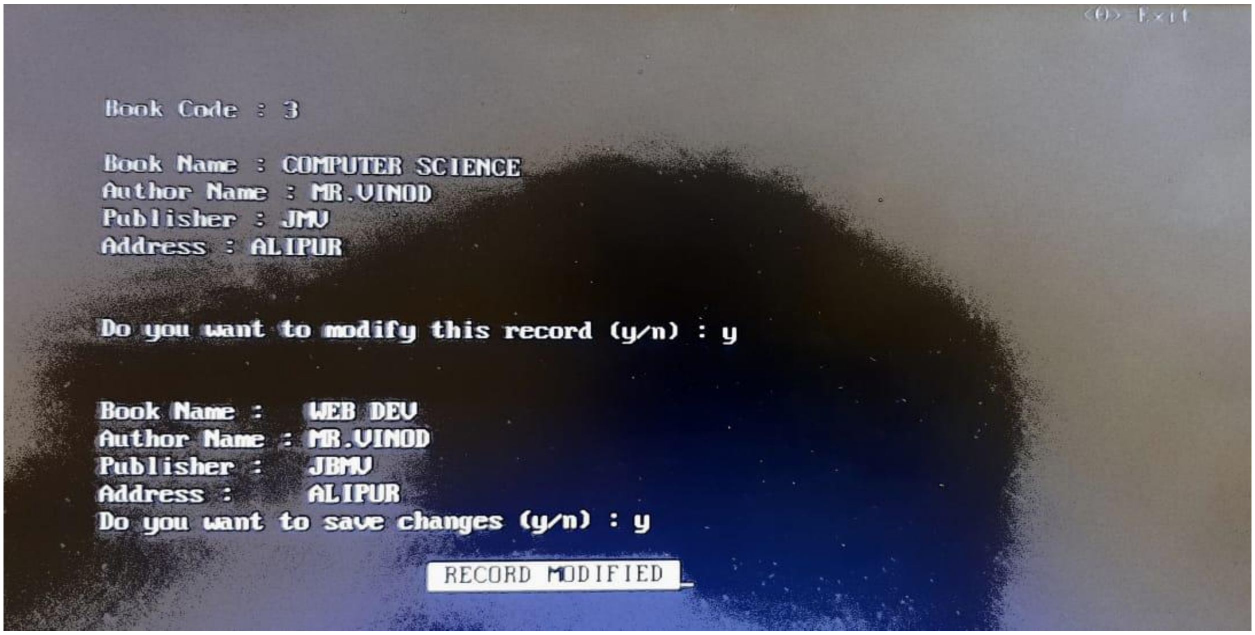
```
                              LIST OF MEMBERS
                             ~~~~~~~~~~~~~~~~~~~
CODE             NAME                         PHONE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 1       YASH TYAGI                  9879763558HOLAMBI KALAN

 3       AYUSH                       8094779898HOLAMBI

 4       ROHAN                       9373986388RAJISTHAN

 5       HARMAN                      5372657730KASDKAWJ






Press any key to continue..._
```



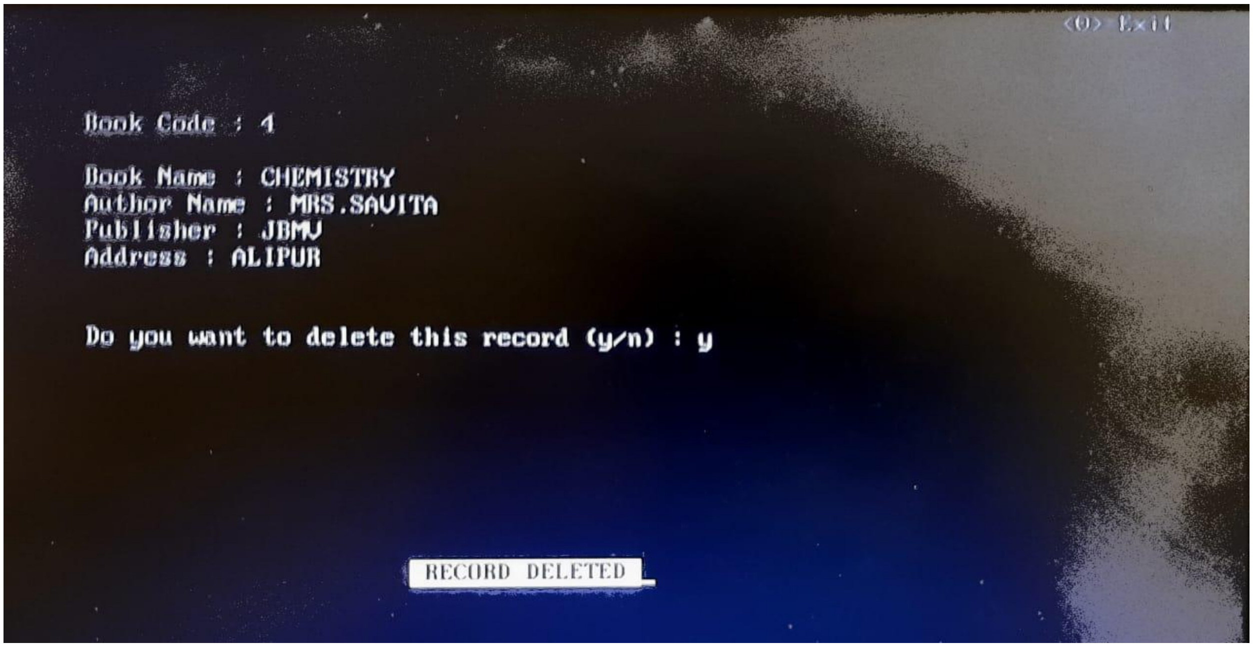
```
                                                                    <0>-Exit

                         ADDITION OF THE BOOKS
                         ~~~~~~~~~~~~~~~~~~~~~

  Code no. 3


  Book Name :    Computer science
  Author Name : Mr.VINOD
  Publisher :    JMV
  Address :      ALIPUR



  Do you want to save (y/n) : y

  Do you want to add more (y/n) : _
```



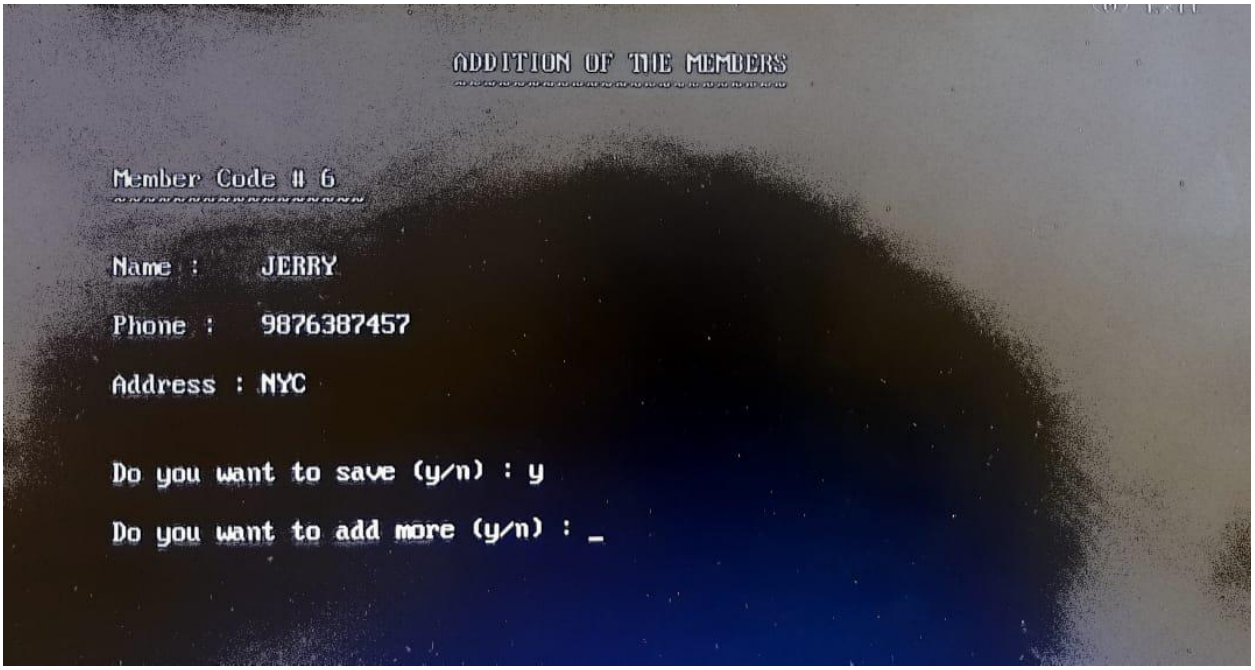
```
                           E D I T B O O K S
                           ~~~~~~~~~~~~~~~~~~


                              1. MODIFY
                              2. DELETE
                              0. EXIT


                          Enter your choice : _
```



```
                                                        <0> Exit

    Book Code : 3

    Book Name : COMPUTER SCIENCE
    Author Name : MR.VINOD
    Publisher : JMV
    Address : ALIPUR


    Do you want to modify this record (y/n) : y


    Book Name :   WEB DEV
    Author Name : MR.VINOD
    Publisher :   JBMV
    Address :     ALIPUR
    Do you want to save changes (y/n) : y
                          RECORD MODIFIED
```



```
                                                        <0> Exit

    Book Code : 4

    Book Name : CHEMISTRY
    Author Name : MRS.SAVITA
    Publisher : JBMV
    Address : ALIPUR


    Do you want to delete this record (y/n) : y







                         RECORD DELETED
```



```
                    ADDITION OF THE MEMBERS
                    ~~~~~~~~~~~~~~~~~~~~~~~


Member Code # 6
~~~~~~~~~~~~~~~

Name :     JERRY

Phone :    9876387457

Address : NYC

Do you want to save (y/n) : y

Do you want to add more (y/n) : _
```