



Library Database Management System

Abdul Raheem 21L-5218

Sarmad Siddique 21L-5225

Abdullah Fayyaz 21I-5208

Main Idea:

1. Ask the user to choose desired operation.
2. Open already made user and books file through functions and read data into structures
3. Get data of new book in struct from user and use function to insert it in already existing file.
4. Similarly, get data and edit the already existing file to delete the desired book.
5. Ask the user the way he wants to search through the books and use overload functions/conditions to search through books.
6. The data is sorted according to user's choice through sort function and displays in a new file.

7. Use embedded search function to find the desired books, and display them in the display function.
8. Edit the contact using edit contact function.
9. Use search function to search through users based on the search books function

Input Variables, arrays and structs:

Global structs:

```

const int c = 1000;

struct BookData
{
    char author[30] = { '\0' };
    char co_author[30] = { '\0' };
    char title[30] = { '\0' };
    char publisher[30] = { '\0' };
    int year_publication = 0;
    char subject[20] = { '\0' };
    int accession_number = 0;
    bool issuable = 1;
    bool issued = 0;
    int date_issuance[3] = { 0 };           // date/month/year
    char issued_toID[10] = { '0','\0'}; 
    int return_date[3] = { 0 };           // date/month/year
};

struct address
{
    int houseno;
    char block[];
    int blockno;
    char town[];
    char city[];

};

struct UserData
{
    char name[30] = { '\0' };
    char father_name[30] = { '\0' };
    char rank[10] = { '\0' };
    char department[30] = { '\0' };
    char ID[10] = { '0' };
    int birthday[3] = { 0 };
    address ad;//Nested struct
    long long int cell_num = { 0 };
};

```

In order to read and modify data from the users and books file two struct datatypes are created.

Array of structures and other variables:

```
int totalbooks = 0, totalusers = 0, choice=0, index=0;
BookData books[c];
UserData users[c];
BookData newbook;

UserData edituser
{
    char name='\0';
    char father='\0';
    char rank='\0';
    char dept='\0';
    char id='\0';
    int dob=0;
    address ad;
    long long int cellno=0;
};
```

The struct datatypes users and books are further used as array of structures and singular variables in order to perform desired operations.

Output Variables:

The output is either changes in the desired file or displaying certain data on the console.

This is done through the same structures `BookData` and `UserData` by further creating singular variables of these datatypes or array of structures.

Function Prototypes and their functionality:

1. Function to read data into array of structures:

```
void Getdata(struct BookData books[c], struct UserData users[c],
            int& totalbooks, int& totalusers);
```

This function will be responsible for reading the book and user data into array of structures and the number of users and books in that file.

Using `ifstream`, we store every piece of data from the files in different arrays in struct books, such that indexes of each array having same value will have data belonging to the same record. Using this we input data from both books.dat and users.dat.

2. Function to insert a book in the existing records:

```
void InsertBook(struct BookData books[c], struct UserData users[c], int& totalbooks);
```

This function will output the data obtained from user in the existing `books.dat` file and assign accession number to the new book and update the data provided by the user.

3. Function to Delete a book from the existing records:

```
void DeleteBook(char title[], struct BookData books[], int totalbooks);
```

This function will be responsible for deleting the book by using search function to search for the title read from user.

First we find the index of the record we want to delete using search function, then we store the whole original `struct BookData books[]` into a temporary array of same struct then we initialize all the values of our original struct array as 0 or null, lastly we copy the temp array into our original struct array but we skip the index of the book or record that we want to delete. Also, we decrement the `total_books` variable by 1. Now, we can update our file by outputting the data from struct `BookData books[]` onto the file.

4. Function to Issue a book:

```
void IssueBook(char title[], char id[], struct BookData books[], int totalbooks);
```

This function will edit the status of the desired book using embedded search function to find the book and update the issuance information.

5. Function to search a book:

```
void SearchBookAuthor(struct BookData books[],char author [],int totalbooks);
void SearchBook_co(struct BookData books[],char coauthor[]int totalbooks);
void SearchBookAno(struct BookData books[],int accession_number,int totalbooks);
void SearchBookSubject(struct BookData books[],char coauthor[]int totalbooks);
```

These functions will search w.r.t the option chosen by the user and use `strcmp()` function to find the desired book.

By using a loop in which we use an if statement with `strcmp() == 0` as a condition, we can find the entry having the same name as given by user.

6. Function to Ascending sort the data:

```
void SortAscend(struct BookData books[], int totalbooks, char filename[],int choice);
```

This function will read the user's choice and filename, sort the existing data according to the choice and output it in the new file.

First of all the user will enter the choice according to it the array will be sorted To sort it first we will create a temporary variable temp, next we will create a counter variable and initialize it to 1, then a while loop will be used which will work till counter is less than length of array.

Also an other for loop will be used in it which will work till I is less than(counter-1)and one if statement will be used in it that(if `strcmp` function will return 1 it will swap the value stored in that index of array).

7. Function to Display books:

```
void DisplayBooks(struct BookData books[],char author_sub[],
                  int totalbooks, int choice);
```

This function will Display all the books written by certain author or subject using search function depending on author's choice.

It will use the stored choice to identify whether the array to be accessed is author or subjects, then after using search function to find the indexes of the books written by the author or of that subject those books will be printed.

8. Function to edit an existing user/contact:

```
void EditContact(int index, struct UserData edituser,  
                 struct UserData users[], int total_users);
```

`edituser` : The contact to be edited is read from user. This struct stores the original data of the index obtained by search function. `edituser` then gets updated based on the users' choice.

This function will take the `edituser` and the user index as it's parameter and swap it with the user data on that index.

9. Function to Search a user:

```
void SearchUserName(struct UserData users[], char name [],int totalusers);  
void SearchUserID(struct UserData users[], char ID[],int totalusers);  
void SearchUserCell(struct UserData users[], long long int cellnumber,int totalusers);
```

These functions will search w.r.t the option chosen by the user and use `strcmp()` function to find the desired user.

Using `strcmp()` function in an if statement in a loop, we find the index of the data given by the user, then by using `SearchBookID` function we can find the books issued to the user and display all that.

Relationship between Functions:

