

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

B.Sc. (1st Sem)

Intro to C Programming Project



Library Management System

Submitted by:

Name
Anjali Chaturvedi
Vibhu Chaudhary
Shreyansh Chaudhary
Kirti Sharma

Enrollment No.
BSG22061
BSG22074
BSG22005
BSG22027

PROJECT OVERVIEW

Aim

To create a command line interface for operating, maintaining and accessing library database to help staff and members.

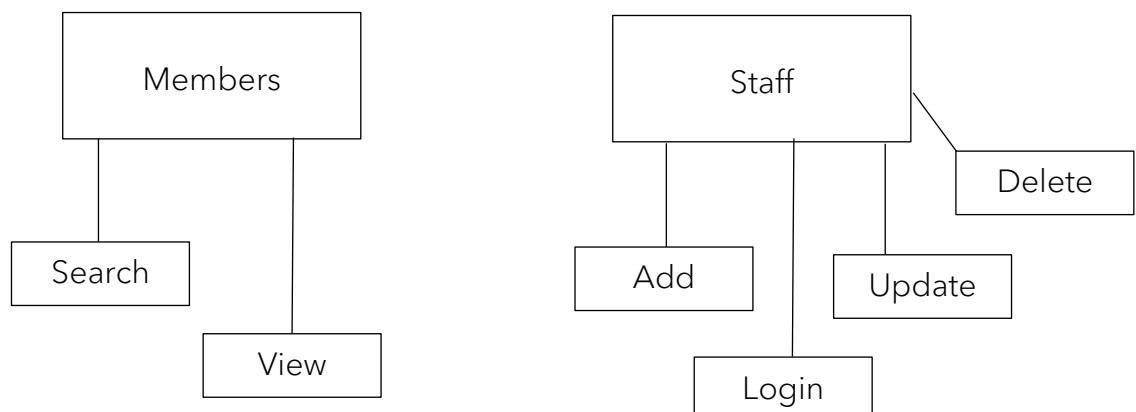
Library System Objectives

- To create a user-friendly command line interface of the library record management system for library staff and members.
- To create a command line interface for library employees to modify and read the records.
- Create a user-friendly "User Authentication System".
- To help library staff organize books by book ID, author name, book type, etc.
- To help members by searching in the library database to check the availability of books
- Make the library system easy to use, maintain and access

Topics of C Used

- Logic analysis
- Requirement analysis
- File Handling
- Loops (do while, for, while)
- Arrays
- Functions
- Data, variables and datatypes

Logic Analysis



Source Code

```
1. #include <stdio.h>
2. #include <time.h>
3. #include <string.h>
4. #define max_year 9999
5. #define min_year 1900
6. #define size_username 30
7. #define size_password 20
8. #define filename "library.bin"
9.
10. // book info input
11. #define bookname 50
12. #define authorname 50
13. #define studentname 50
14. #define studentaddress 300
15. #define size_fileheader sizeof(sFileHeader)
16. //structure to store date
17. // genre, availability, book id
18.
19. int main()
20. {
21.     defaultLogin();
22.     welcomeMessage();
23.     login();
24.     return 0;
25. }
26.
27. // date structure
28. typedef struct
29. {
30.     int yyyy;
31.     int mm;
32.     int dd;
33. } Date;
34.
35. // login structure
36. typedef struct
37. {
38.     char username[size_username];
39.     char password[size_password];
40. } sFileHeader;
41.
42. // book info
43. typedef struct
44. {
45.     int books_id;
46.     char bookName[bookname];
47.     char authorName[authorname];
48.     char studentName[studentname];
49.     char studentAddr[studentaddress];
50.     Date bookIssueDate;
51. } s_BooksInfo;
52.
53. // login message
54. void printMessageCenter(const char* message)
55. {
56.     int len =0;
57.     int pos = 0;
58.     len = (78 - strlen(message))/2;
59.     printf("\t\t\t\t");
60.     for(pos =0 ; pos < len ; pos++)
61.     {
62.         printf(" ");
63.     }
64.     printf("%s",message);
65. }
66.
67. // initial message
```

```
68. void headMessage(const char *message)
69. {
70.     system("cls");
71.     printf("\n\t\t\t\t\tLibrary Management System\t\t\t\t");
72.     printf("\n\t\t\t\t\t-----\n\n");
73.     printMessageCenter(message);
74.     printf("\n\t\t\t\t\t-----\n\n");
75. }
76.
77. // welcome
78. void welcomeMessage()
79. {
80.     headMessage("C Project");
81.     printf("\n\n\n\t\t\t Enter any key to continue.....");
82.     getch();
83. }
84.
85. // check if name is valid
86. int isValidName(const char *name)
87. {
88.     int validName = 1;
89.     int len = 0;
90.     int index = 0;
91.     len = strlen(name);
92.     for(index = 0; index < len ; ++index)
93.     {
94.         if(!isalpha(name[index])) && (name[index] != '\n') && (name[index] != ' ')
95.         {
96.             validName = 0;
97.             break;
98.         }
99.     }
100.    return validName;
101. }
102.
103. //check for leap year-use in valid date
104. int IsLeapYear(int year)
105. {
106.    return (((year % 4 == 0) &&
107.            (year % 100 != 0)) ||
108.           (year % 400 == 0));
109. }
110.
111. int isValidDate(Date *validDate)
112. {
113.    if (validDate->yyyy > max_year ||
114        validDate->yyyy < min_year)
115.        return 0;
116.    if (validDate->mm < 1 || validDate->mm > 12)
117.        return 0;
118.    if (validDate->dd < 1 || validDate->dd > 31)
119.        return 0;
120.    if (validDate->mm == 2)
121.    {
122.        if (IsLeapYear(validDate->yyyy)) //leap year is used to determine the number of
months in feb
123.            return (validDate->dd <= 29);
124.        else
125.            return (validDate->dd <= 28);
126.    }
127.    if (validDate->mm == 4 || validDate->mm == 6 ||
128.        validDate->mm == 9 || validDate->mm == 11)
129.        return (validDate->dd <= 30);
130.    return 1;
131. }
132.
133. void addBookInDataBase()
134. {
```

```

135.     int days;
136.     s_BooksInfo addBookInfoInDataBase = {0};
137.     FILE *fp = NULL;
138.     int status = 0;
139.     fp = fopen(filename, "ab+");
140.     if(fp == NULL)
141.     {
142.         printf("File is not opened\n");
143.         exit(1);
144.     }
145.     headMessage("ADD NEW BOOKS");
146.     printf("\n\n\t\t\tENTER YOUR DETAILS BELOW:");
147.     printf("\n\t\t\t-----\n");
148.     printf("\n\t\t\tBook ID NO  = ");
149.     fflush(stdin);
150.     scanf("%u",&addBookInfoInDataBase.books_id);
151.     do
152.     {
153.         printf("\n\t\t\tBook Name  = ");
154.         fflush(stdin);
155.         fgets(addBookInfoInDataBase.bookName,bookname,stdin);
156.         status = isNameValid(addBookInfoInDataBase.bookName);
157.         if (!status)
158.         {
159.             printf("\n\t\t\tName contain invalid character. Please enter again.");
160.         }
161.     }
162.     while(!status);
163.     do
164.     {
165.         printf("\n\t\t\tAuthor Name  = ");
166.         fflush(stdin);
167.         fgets(addBookInfoInDataBase.authorName,authorname,stdin);
168.         status = isNameValid(addBookInfoInDataBase.authorName);
169.         if (!status)
170.         {
171.             printf("\n\t\t\tName contain invalid character. Please enter again.");
172.         }
173.     }
174.     while(!status);
175.     do
176.     {
177.         printf("\n\t\t\tStudent Name  = ");
178.         fflush(stdin);
179.         fgets(addBookInfoInDataBase.studentName,studentname,stdin);
180.         status = isNameValid(addBookInfoInDataBase.studentName);
181.         if (!status)
182.         {
183.             printf("\n\t\t\tName contain invalid character. Please enter again.");
184.         }
185.     }
186.     while(!status);
187.     do
188.     {
189.         printf("\n\t\t\tEnter date in format (day/month/year): ");
190.         scanf("%d/%d/%d",&addBookInfoInDataBase.bookIssueDate.dd,&addBookInfoInDataBase.bookIssueDate.mm
,&addBookInfoInDataBase.bookIssueDate.yyyy);
191.         status = isValidDate(&addBookInfoInDataBase.bookIssueDate);
192.         if (!status)
193.         {
194.             printf("\n\t\t\tPlease enter a valid date.\n");
195.         }
196.     }
197.     while(!status);
198.     fwrite(&addBookInfoInDataBase,sizeof(addBookInfoInDataBase), 1, fp);
199.     fclose(fp);
200. }
201.

```

```

202. void searchBooks()
203. {
204.     int found = 0;
205.     char bookName[bookname] = {0};
206.     s_BooksInfo addBookInfoInDataBase = {0};
207.     FILE *fp = NULL;
208.     int status = 0;
209.     fp = fopen(filename,"rb");
210.     if(fp == NULL)
211.     {
212.         printf("\n\t\t\tFile is not opened\n");
213.         exit(1);
214.     }
215.     headMessage("SEARCH BOOKS");
216.     if (fseek(fp,size_fileheader,SEEK_SET) != 0)
217.     {
218.         fclose(fp);
219.         printf("\n\t\t\tFacing issue while reading file\n");
220.         exit(1);
221.     }
222.     printf("\n\n\t\t\tEnter Book Name to search:");
223.     fflush(stdin);
224.     fgets(bookName,bookname,stdin);
225.     while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
226.     {
227.         if(!strcmp(addBookInfoInDataBase.bookName, bookName))
228.         {
229.             found = 1;
230.             break;
231.         }
232.     }
233.     if(found)
234.     {
235.         printf("\n\t\t\tBook id = %u\n",addBookInfoInDataBase.books_id);
236.         printf("\t\t\tBook name = %s",addBookInfoInDataBase.bookName);
237.         printf("\t\t\tBook authorName = %s",addBookInfoInDataBase.authorName);
238.         printf("\t\t\tBook issue date(day/month/year) =
239.             (%d/%d/%d)",addBookInfoInDataBase.bookIssueDate.dd,
240.                 addBookInfoInDataBase.bookIssueDate.mm,
241.                 addBookInfoInDataBase.bookIssueDate.yyyy);
242.     }
243.     else
244.     {
245.         printf("\n\t\t\tNo Record");
246.     }
247.     fclose(fp);
248.     printf("\n\n\n\t\t\tPress any key to go to main menu.....");
249.     getchar();
250. // v books function
251. void viewBooks()
252. {
253.     int found = 0;
254.     char bookName[bookname] = {0};
255.     s_BooksInfo addBookInfoInDataBase = {0};
256.     FILE *fp = NULL;
257.     int status = 0;
258.     unsigned int countBook = 1;
259.     headMessage("VIEW BOOKS DETAILS");
260.     fp = fopen(filename,"rb");
261.     if(fp == NULL)
262.     {
263.         printf("File is not opened\n");
264.         exit(1);
265.     }
266.     if (fseek(fp,size_fileheader,SEEK_SET) != 0)
267.     {
268.         fclose(fp);
269.         printf("Facing issue while reading file\n");
270.         exit(1);

```

```

270.     }
271.     while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
272.     {
273.         printf("\n\t\t\tBook Count = %d\n\n", countBook);
274.         printf("\t\t\tBook id = %u", addBookInfoInDataBase.books_id);
275.         printf("\n\t\t\tBook name = %s", addBookInfoInDataBase.bookName);
276.         printf("\t\t\tBook authorName = %s", addBookInfoInDataBase.authorName);
277.         printf("\t\t\tBook issue date(day/month/year) =
278.         (%d/%d/%d)\n\n", addBookInfoInDataBase.bookIssueDate.dd,
279.         addBookInfoInDataBase.bookIssueDate.mm,
280.         addBookInfoInDataBase.bookIssueDate.yyyy);
281.         found = 1;
282.         ++countBook;
283.     }
284.     fclose(fp);
285.     if(!found)
286.     {
287.         printf("\n\t\t\tNo Record");
288.     }
289.     printf("\n\n\t\t\tPress any key to go to main menu.....");
290.     fflush(stdin);
291.     getchar();
292. }
293. // delete function
294. void deleteBooks()
295. {
296.     int found = 0;
297.     int bookDelete = 0;
298.     sFileHeader fileHeaderInfo = {0};
299.     char bookName[bookname] = {0};
300.     s_BooksInfo addBookInfoInDataBase = {0};
301.     FILE *fp = NULL;
302.     FILE *tmpFp = NULL;
303.     int status = 0;
304.     headMessage("Delete Books Details");
305.     fp = fopen(filename, "rb");
306.     if(fp == NULL)
307.     {
308.         printf("File is not opened\n");
309.         exit(1);
310.     }
311.     tmpFp = fopen("tmp.bin", "wb");
312.     if(tmpFp == NULL)
313.     {
314.         fclose(fp);
315.         printf("File is not opened\n");
316.         exit(1);
317.     }
318.     fread (&fileHeaderInfo, size_fileheader, 1, fp);
319.     fwrite(&fileHeaderInfo, size_fileheader, 1, tmpFp);
320.     printf("\n\t\t\tEnter Book ID NO. for delete:");
321.     scanf("%d", &bookDelete);
322.     while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
323.     {
324.         if(addBookInfoInDataBase.books_id != bookDelete)
325.         {
326.             fwrite(&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, tmpFp);
327.         }
328.         else
329.         {
330.             found = 1;
331.         }
332.     }
333.     (found)? printf("\n\t\t\tRecord deleted successfully....."):printf("\n\t\t\tRecord not
334.     found");
335.     fclose(fp);
336.     fclose(tmpFp);
337.     remove(filename);
338.     rename("tmp.bin", filename);
339. }

```

```

337.
338. void updateCredential(void)
339. {
340.     sFileHeader fileHeaderInfo = {0};
341.     FILE *fp = NULL;
342.     unsigned char userName[size_username] = {0};
343.     unsigned char password[size_password] = {0};
344.     headMessage("Update Credential");
345.     fp = fopen(filename, "rb+");
346.     if(fp == NULL)
347.     {
348.         printf("File is not opened\n");
349.         exit(1);
350.     }
351.     fread (&fileHeaderInfo, size_fileheader, 1, fp);
352.     if (fseek(fp, 0, SEEK_SET) != 0)
353.     {
354.         fclose(fp);
355.         printf("\n\t\t\tFacing issue while updating password\n");
356.         exit(1);
357.     }
358.     printf("\n\n\t\t\tNew Username:");
359.     fflush(stdin);
360.     fgets(userName, size_username, stdin);
361.     printf("\n\n\t\t\tNew Password:");
362.     fflush(stdin);
363.     fgets(password, size_password, stdin);
364.     strncpy(fileHeaderInfo.username, userName, sizeof(userName));
365.     strncpy(fileHeaderInfo.password, password, sizeof(password));
366.     fwrite(&fileHeaderInfo, size_fileheader, 1, fp);
367.     fclose(fp);
368.     printf("\n\t\t\tYour Password has been changed successfully");
369.     printf("\n\t\t\t\tLogin Again:");
370.     fflush(stdin);
371.     getchar();
372.     exit(1);
373. }
374.
375. void menu()
376. {
377.     int choice = 0;
378.     do
379.     {
380.         headMessage("MAIN MENU");
381.         printf("\n\n\n\t\t\t1.Add Books");
382.         printf("\n\t\t\t2.Search Books");
383.         printf("\n\t\t\t3.View Books");
384.         printf("\n\t\t\t4.Delete Book");
385.         printf("\n\t\t\t5.Update Password");
386.         printf("\n\t\t\t0.Exit");
387.         printf("\n\n\n\t\t\tEnter choice => ");
388.         scanf("%d", &choice);
389.         switch(choice)
390.         {
391.             case 1:
392.                 addBookInDataBase();
393.                 break;
394.             case 2:
395.                 searchBooks();
396.                 break;
397.             case 3:
398.                 viewBooks();
399.                 break;
400.             case 4:
401.                 deleteBooks();
402.                 break;
403.             case 5:
404.                 updateCredential();
405.                 break;
406.             case 0:

```



```

407.         printf("\n\n\n\t\t\tThank you!!!\n\n\n\n");
408.         exit(1);
409.         break;
410.     default:
411.         printf("\n\n\n\t\t\tINVALID INPUT!!! Try again...");
412.     }
413. }
414. while(choice!=0);
415. }
416.
417. //login password
418. void login()
419. {
420.     unsigned char userName[size_username] = {0};
421.     unsigned char password[size_password] = {0};
422.     int L=0;
423.     sFileHeader fileHeaderInfo = {0};
424.     FILE *fp = NULL;
425.     headMessage("Login");
426.     fp = fopen(filename,"rb");
427.     if(fp == NULL)
428.     {
429.         printf("File is not opened\n");
430.         exit(1);
431.     }
432.     fread (&fileHeaderInfo,size_fileheader, 1, fp);
433.     fclose(fp);
434.     do
435.     {
436.         printf("\n\n\n\t\t\tUsername:");
437.         fgets(userName,size_username,stdin);
438.         printf("\n\t\t\t\tPassword:");
439.         fgets(password,size_password,stdin);
440.         if((!strcmp(userName,fileHeaderInfo.username)) &&
(!strcmp(password,fileHeaderInfo.password)))
441.         {
442.             menu();
443.         }
444.         else
445.         {
446.             printf("\t\t\t\tLogin Failed Enter Again Username & Password\n\n");
447.             L++;
448.         }
449.     }
450.     while(L<=3);
451.     if(L>3)
452.     {
453.         headMessage("Login Failed");
454.         printf("\t\t\t\tSorry,Unknown User.");
455.         getch();
456.         system("cls");
457.     }
458. }
459.
460. int isFileExists(const char *path)
461. {
462.     FILE *fp = fopen(path, "rb");
463.     int status = 0;
464.     if (fp != NULL)
465.     {
466.         status = 1;
467.         fclose(fp);
468.     }
469.     return status;
470. }
471.
472. void defaultLogin()
473. {
474.     FILE *fp = NULL;
475.     int status = 0;

```

```
476.     const char defaultUsername[] = "jiit\n";
477.     const char defaultPassword[] = "jiit\n";
478.     sFileHeader fileHeaderInfo = {0};
479.     status = isFileExists(filename);
480.     if(!status)
481.     {
482.         fp = fopen(filename, "wb");
483.         if(fp != NULL)
484.         {
485.             strncpy(fileHeaderInfo.password, defaultPassword, sizeof(defaultPassword));
486.             strncpy(fileHeaderInfo.username, defaultUsername, sizeof(defaultUsername));
487.             fwrite(&fileHeaderInfo, size_fileheader, 1, fp);
488.             fclose(fp);
489.         }
490.     }
491. }
492.
```

Final Output

```
Library Management System
-----
C Project
-----

Enter any key to continue.....
```

```
Library Management System
-----
Login
-----

Username:jiit
Password:jiit
```

```
Library Management System
-----
MAIN MENU
-----

1.Add Books
2.Search Books
3.View Books
4.Delete Book
5.Update Password
0.Exit

Enter choice =>
```

```
Library Management System
-----
ADD NEW BOOKS
-----

ENTER YOUR DETAILS BELOW:
-----

Book ID NO =
```

```
Library Management System
-----
SEARCH BOOKS
-----

Enter Book Name to search:Design Patterns

Book id = 2
Book name = Design Patterns
Book authorName = Erich Gamma
Book issue date(day/month/year) = (10/12/2022)

Press any key to go to main menu.....
```

```
Library Management System
-----
VIEW BOOKS DETAILS
-----
Book Count = 1

Book id = 1
Book name = Introduction to Algorithms
Book authorName = Thomas Cormen
Book issue date(day/month/year) = (30/12/2022)

Book Count = 2

Book id = 2
Book name = Design Patterns
Book authorName = Erich Gamma
Book issue date(day/month/year) = (10/12/2022)

Book Count = 3

Book id = 3
Book name = The Pragmatic Programmer
Book authorName = Andrew Hunt
Book issue date(day/month/year) = (12/10/2022)
```

```
Library Management System
-----
Delete Books Details
-----
Enter Book ID NO. for delete:1
```

```
Library Management System
-----
Update Credential
-----
New Username:
```