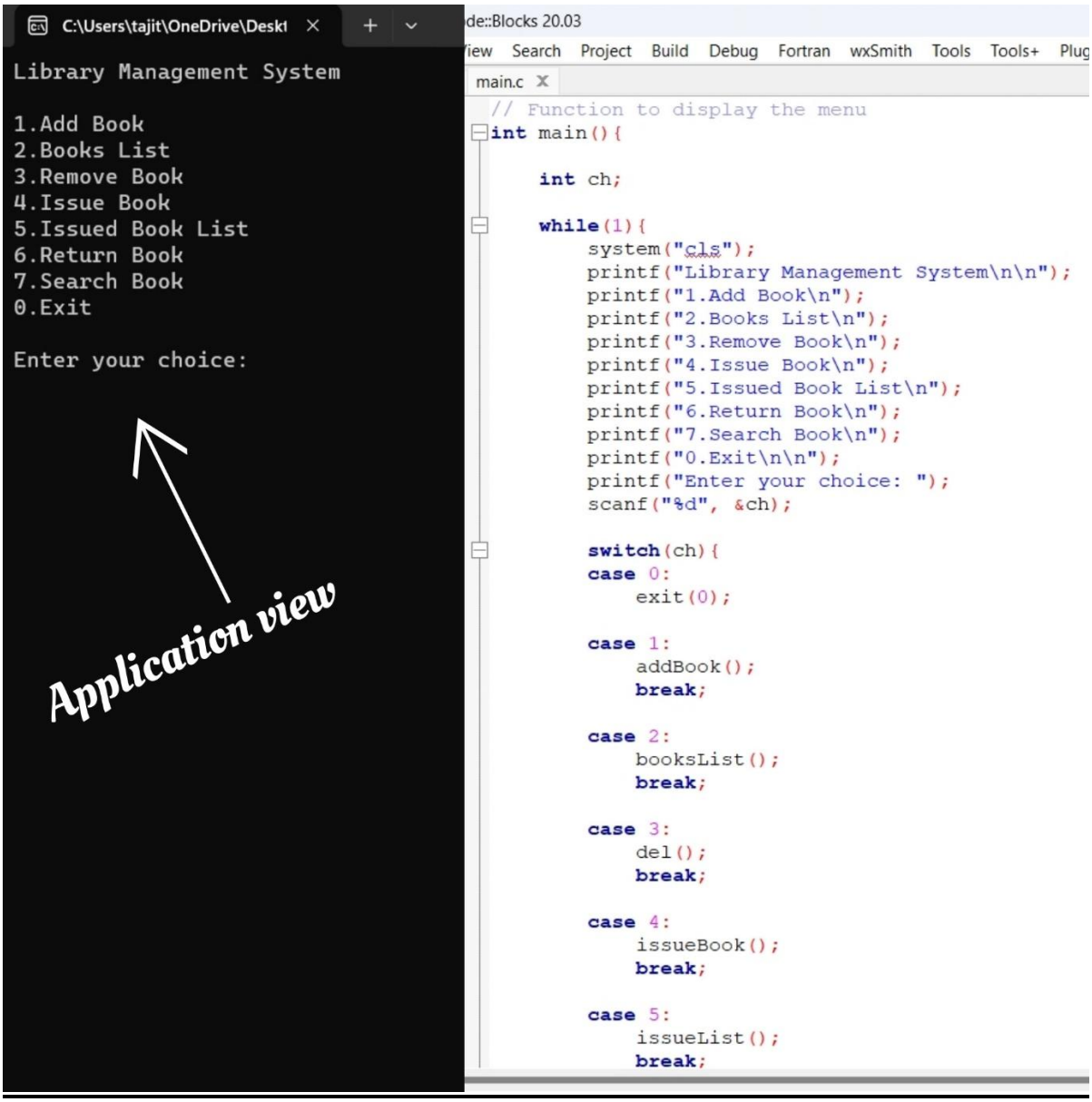
LIBRARY MANAGEMENT SYSTEM

Project Report by Tajith Rashidha

Table of Contents

1. **Fulfilled requirements with an explanation and a screenshot of the codes.**

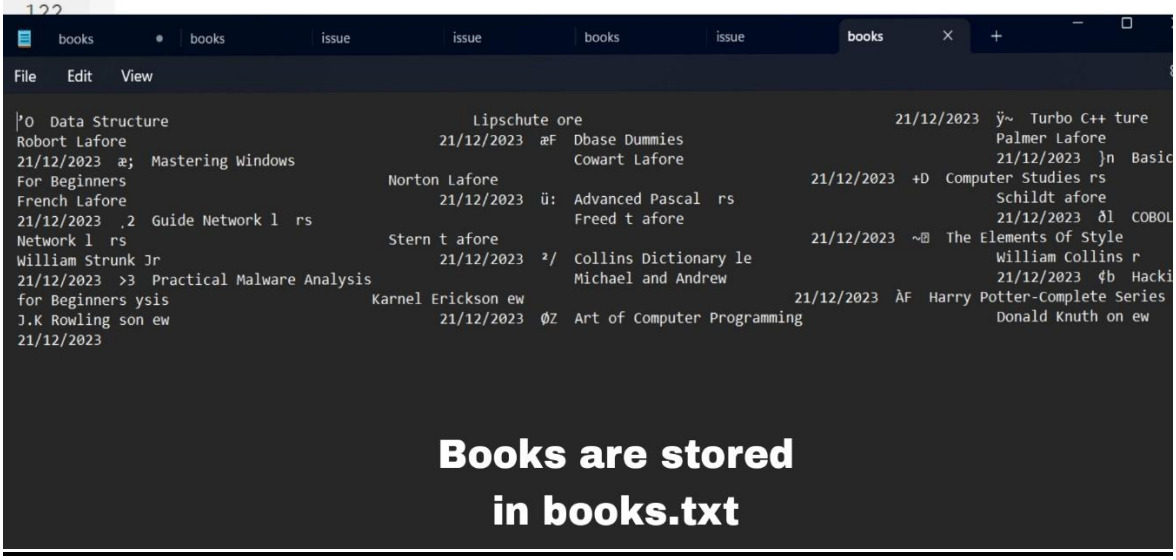
1.1 Main Menu (main function)



- The main function displays a menu for a library management system.
- It uses a 'switch' statement to handle different user choices.
- It incorporates a perpetual loop, allowing the user to perform multiple operations until choosing to exit.

1.2 Adding a Book (addBook function)

```c
// Function to add a book to the library
void addBook(){
    char myDate[12];
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon+1, tm.tm_year + 1900);
    strcpy(b.date, myDate);

    fp = fopen("books.txt", "ab");

    printf("Enter book id: ");
    scanf("%d", &b.id);

    printf("Enter book name: ");
    fflush(stdin);
    gets(b.bookName);

    printf("Enter author name: ");
    fflush(stdin);
    gets(b.authorName);

    printf("Book Added Successfully");

    fwrite(&b, sizeof(b), 1, fp);
    fclose(fp);
}
```

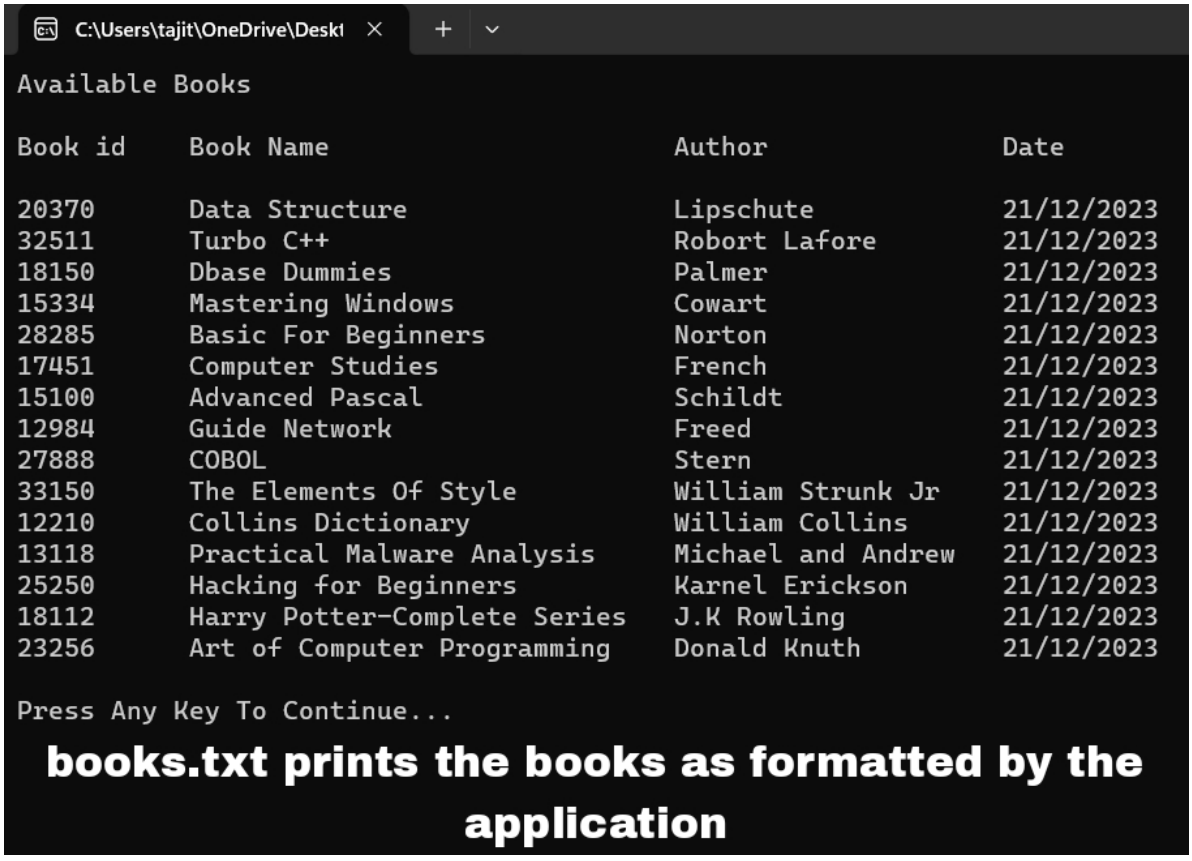


- To add a book to the library, use the ‘addBook’ function.
- It captures current date and time and associates it with the book.
- Book details are entered by the user and saved in a binary file called ("books.txt").
- Uses ‘sprintf’ and ‘strcpy’ functions to format and copy date information.

1.3 Displaying Books (booksList function)

```
123
124   // Function to display all books in the library
125   void booksList(){
126
127       system("cls");
128       printf("Available Books\n\n");
129       printf("%-10s %-30s %-20s %s\n\n", "Book id", "Book Name", "Author", "Date");
130
131       fp = fopen("books.txt", "rb");
132       while(fread(&b, sizeof(b), 1, fp) == 1){
133           printf("%-10d %-30s %-20s %s\n", b.id, b.bookName, b.authorName, b.date);
134       }
135
136       fclose(fp);
137   }
```

- The 'booksList' function displays all available books.
- Reads book details from the "books.txt" file and prints them in a formatted manner.



1.4 Removing a Book (del function)

```
138  // Function to remove a book from the library
139  void del() {
140      int id, f = 0;
141      system("cls");
142      printf("Remove Books\n\n");
143      printf("Enter Book id to remove: ");
144      scanf("%d", &id);
145
146      FILE *ft;
147
148      fp = fopen("books.txt", "rb");
149      ft = fopen("temp.txt", "wb");
150
151      while (fread(&b, sizeof(b), 1, fp) == 1) {
152          if (id == b.id) {
153              f = 1;
154          } else {
155              fwrite(&b, sizeof(b), 1, ft);
156          }
157      }
158
159      if (f == 1) {
160          printf("\n\nDeleted Successfully.");
161      } else {
162          printf("\n\nRecord Not Found !");
163      }
164
165      fclose(fp);
166      fclose(ft);
167
168      remove("books.txt");
169      rename("temp.txt", "books.txt");
170  }
```

- A book is removed by the ‘del’ function in response to user input.
- It reads books from "books.txt," and doesn't write them to a temporary file if the given book ID is located.
- The temporary file is used in place of the original file following processing.

1.5 Issuing a Book (issueBook function)

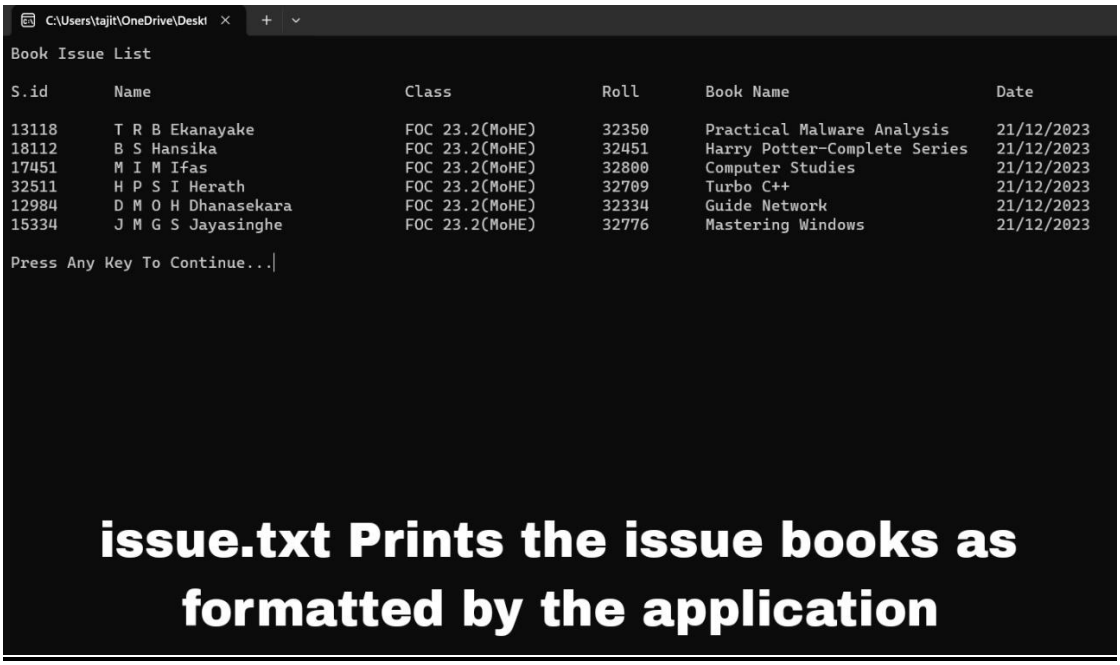
```
// Function to display issue book
void issueBook(){

    char myDate[12];
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon+1, tm.tm_year + 1900);
    strcpy(s.date, myDate);

    int f=0;

    system("cls");
    printf("Issue Books\n\n");

    printf("Enter Book id to issue: ");
    scanf("%d", &s.id);

    fp = fopen("books.txt", "rb");

    while(fread(&b, sizeof(b), 1, fp) == 1){
        if(b.id == s.id){
            strcpy(s.bookName, b.bookName);
            f=1;
            break;
        }
    }

    if(f==0){
        printf("No book found with this id\n");
        printf("Please try again...\n\n");
        return;
    }

    fp = fopen("issue.txt", "ab");

    printf("Enter Student Name: ");
    fflush(stdin);
    gets(s.sName);

    printf("Enter Student Class: ");
    fflush(stdin);
    gets(s.sClass);

    printf("Enter Student Roll: ");
    scanf("%d", &s.sRoll);

    printf("Book Issued Successfully\n\n");

    fwrite(&s, sizeof(s), 1, fp);
    fclose(fp);
}
```

- The ‘issueBook’ function issues a book to a student.
- It reads the book details from "books.txt," captures the current date, and records the transaction in "issue.txt."

1.6 Displaying Issued Books (issueList function)

```
222
223  // Function to display issue list
224  void issueList(){
225      system("cls");
226      printf("Book Issue List\n\n");
227
228      printf("%-10s %-30s %-20s %-10s %-30s %s\n\n", "S.id", "Name", "Class", "Roll", "Book Name", "Date");
229
230      fp = fopen("issue.txt", "rb");
231      while(fread(&s, sizeof(s), 1, fp) == 1){
232          printf("%-10d %-30s %-20s %-10d %-30s %s\n", s.id, s.sName, s.sClass, s.sRoll, s.bookName, s.date);
233      }
234
235      fclose(fp);
236  }
237
```

- A list of books that have been issued, along with student information and the issue date, is shown by the 'issueList' function.
- reads information from "issue.txt" and outputs structured data.



1.7 Returning a Book (returnBook function)

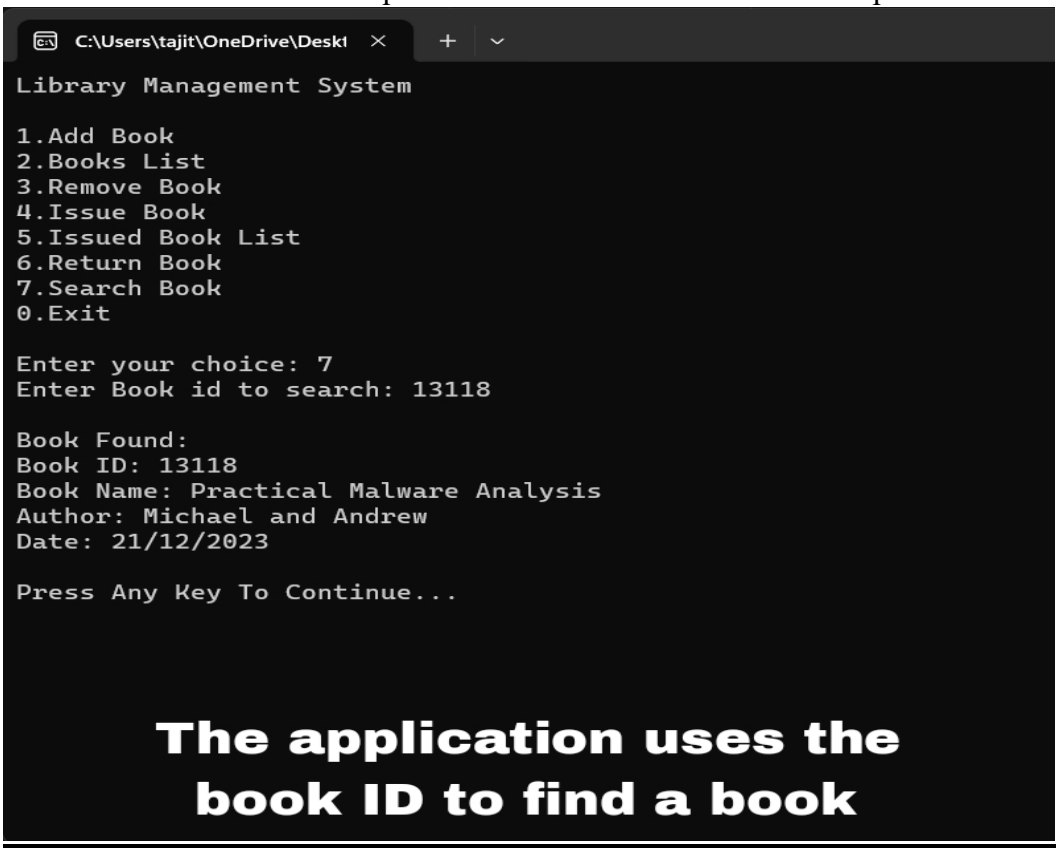
```
238   // Function to return a book
239   void returnBook() {
240       int id, f = 0;
241       system("cls");
242       printf("Return Books\n\n");
243       printf("Enter Book id to return: ");
244       scanf("%d", &id);
245
246       FILE *ft;
247
248       fp = fopen("issue.txt", "rb");
249       ft = fopen("temp_issue.txt", "wb");
250
251       while (fread(&s, sizeof(s), 1, fp) == 1) {
252           if (id == s.id) {
253               f = 1;
254               printf("\n\nBook Returned Successfully.");
255           } else {
256               fwrite(&s, sizeof(s), 1, ft);
257           }
258       }
259
260       if (f == 0) {
261           printf("\n\nBook ID Not Found !");
262       }
263
264       fclose(fp);
265       fclose(ft);
266
267       remove("issue.txt");
268       rename("temp_issue.txt", "issue.txt");
269   }
270
```

- The' returnBook' function processes the return of a book.
- It reads from "issue.txt," and it doesn't write to a temporary file if the given book ID is located.
- The temporary file is used in place of the original file following processing.

1.8 Searching for a Book (searchBook function)

```
271  // Function to search book
272  void searchBook() {
273      int id, found = 0;
274
275      printf("Enter Book id to search: ");
276      scanf("%d", &id);
277
278      fp = fopen("books.txt", "rb");
279
280      while (fread(&b, sizeof(b), 1, fp) == 1) {
281          if (id == b.id) {
282              found = 1;
283              printf("\nBook Found:\n");
284              printf("Book ID: %d\nBook Name: %s\nAuthor: %s\nDate: %s\n", b.id, b.bookName, b.authorName, b.date);
285              break;
286          }
287      }
288
289      fclose(fp);
290
291      if (!found) {
292          printf("\nBook not found with ID: %d\n", id);
293      }
294  }
295
```

- The 'searchBook' function allows the user to search for a book by its ID.
- It reads from "books.txt" and prints the details if the book with the specified ID is found.



2. Where and How to Use Best Practices

2.1 Modularization

- The code is modularized using functions, which improves readability and maintainability. Each function serves a specific function.

2.2 Switch Statement

- Using a ‘switch’ statement instead of several if statements improve readability and maintainability in the main menu.

2.3 Best Practices for File Handling

- File operations (‘fopen’, ‘fclose’, ‘fwrite’, ‘fread’) are enclosed in appropriate blocks and error handling is minimal but present, ensuring robust file handling.

2.4 Time Handling

- To ensure correct timestamping, it's a good idea to capture and format the current date using ‘time.h’.

2.5 User Input Handling

- By properly handling and validating user input, the possibility of unexpected behavior is decreased.

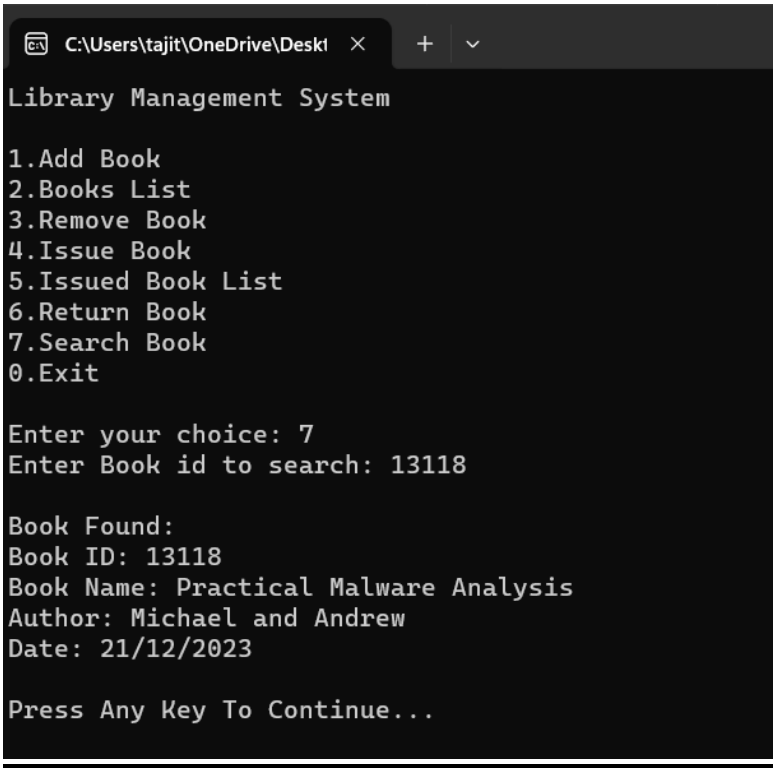
2.6 Formatted Output

- Readability is improved when book and issue details are displayed using prepared output.

3. Additional Features Added

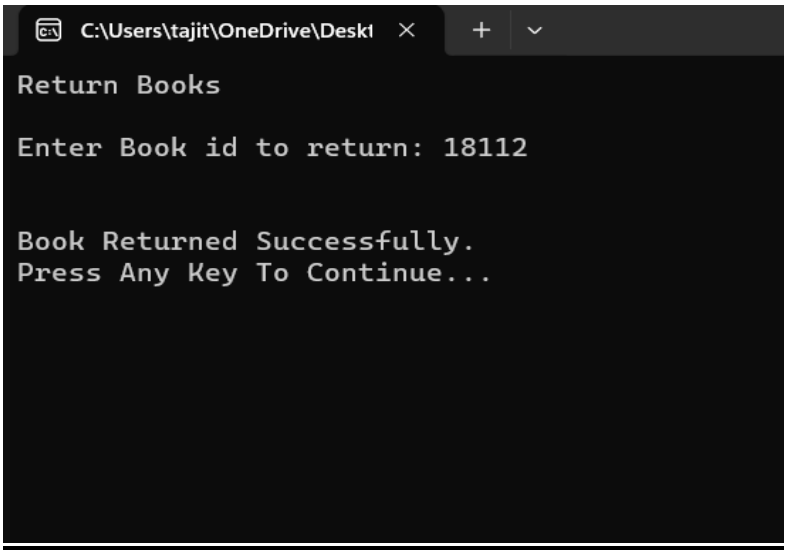
3.1 Search Book (searchfunction)

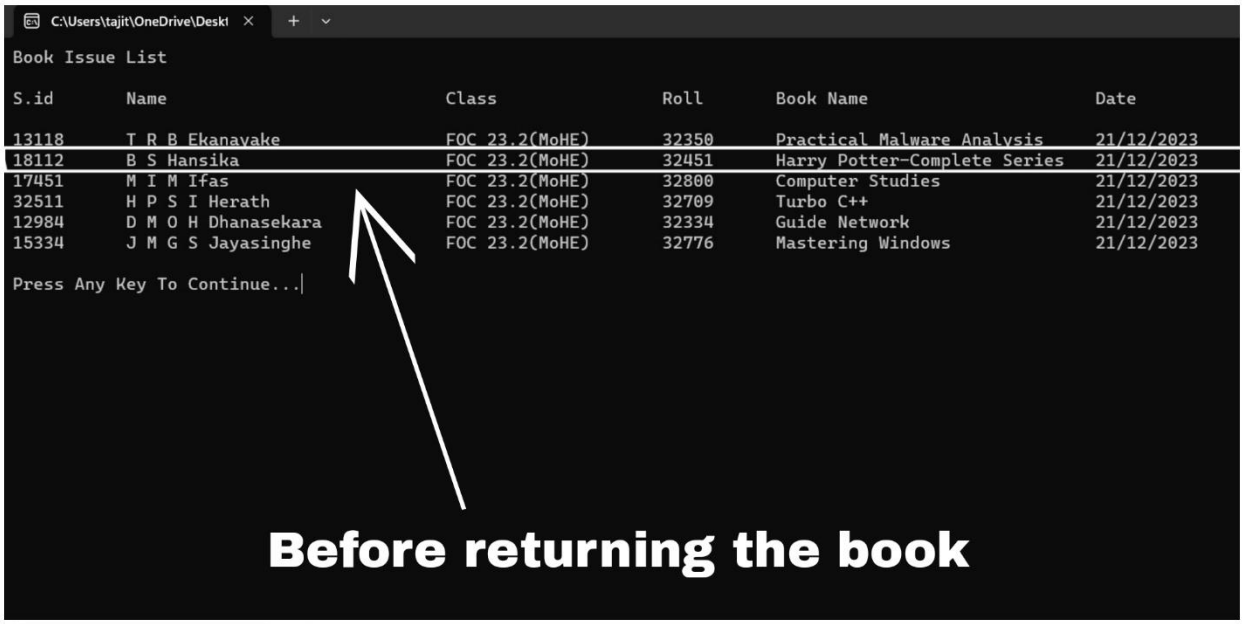
- Allows users to search for a book by its ID, providing details if found.

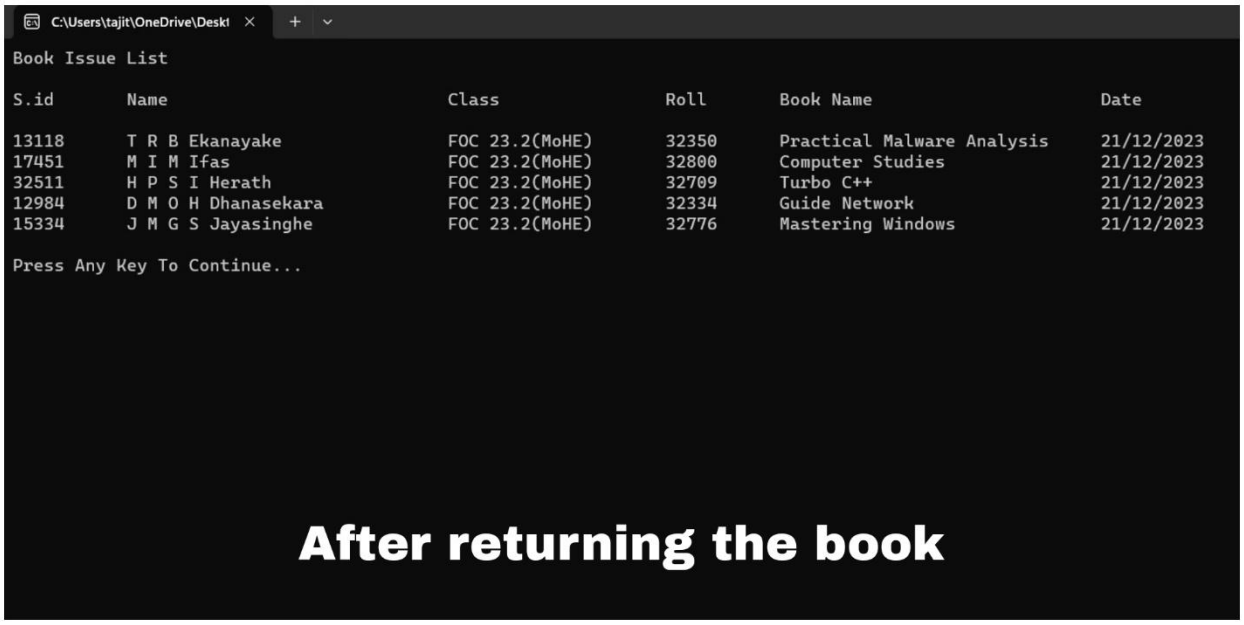


3.2 Return Book (returnBook function)

- Enables users to return a book, updating the issue records.



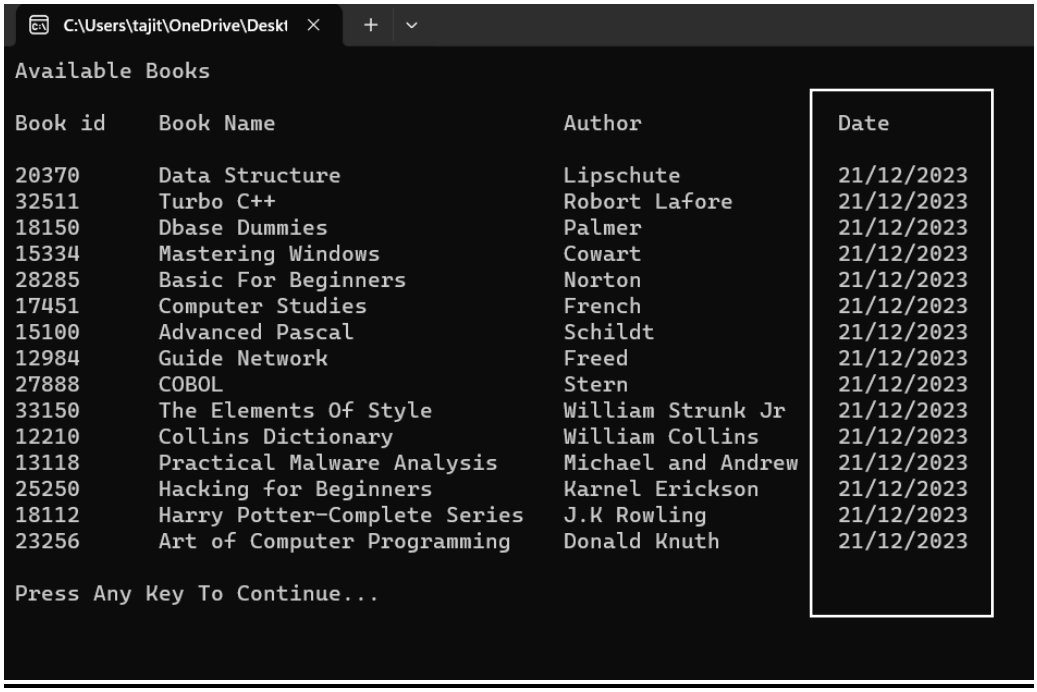




- Following the book's return, Book id 18112 is automatically removed from the list of issued books.

3.3 Timestamping

- Timestamps are added for both book addition and book issuance, providing a record of when actions were performed.



3.4 File Handling

- The new features are handled appropriately, upholding consistency and dependability.

3.5 Validation of User Input

- Data integrity is ensured by applying input validation to stop invalid entries.

Reference table

#	Author(s)	Year	Title	Publisher/URL	In-text Reference
1	Stroustrup, B.	2013	Programming: Principles and Practice Using C++	Addison-Wesley	(Stroustrup, 2013)
2	Kernighan, B. W. & Ritchie, D. M.	1988	The C Programming Language	Prentice Hall	(Kernighan & Ritchie, 1988)
3	cplusplus.com	n.d.	C Standard Library Documentation	cplusplus.com	(cplusplus.com)

Reference List

- Stroustrup, B. (2013) *Programming: Principles and Practice Using C++.* Addison-Wesley.
- Kernighan, B. W., & Ritchie, D. M. (1988) *The C Programming Language.* Prentice Hall.
- cplusplus.com (n.d.) *C Standard Library Documentation.* [Online] Available at: cplusplus.com