Table of Contents

Library Management System

Introduction and Assumptions

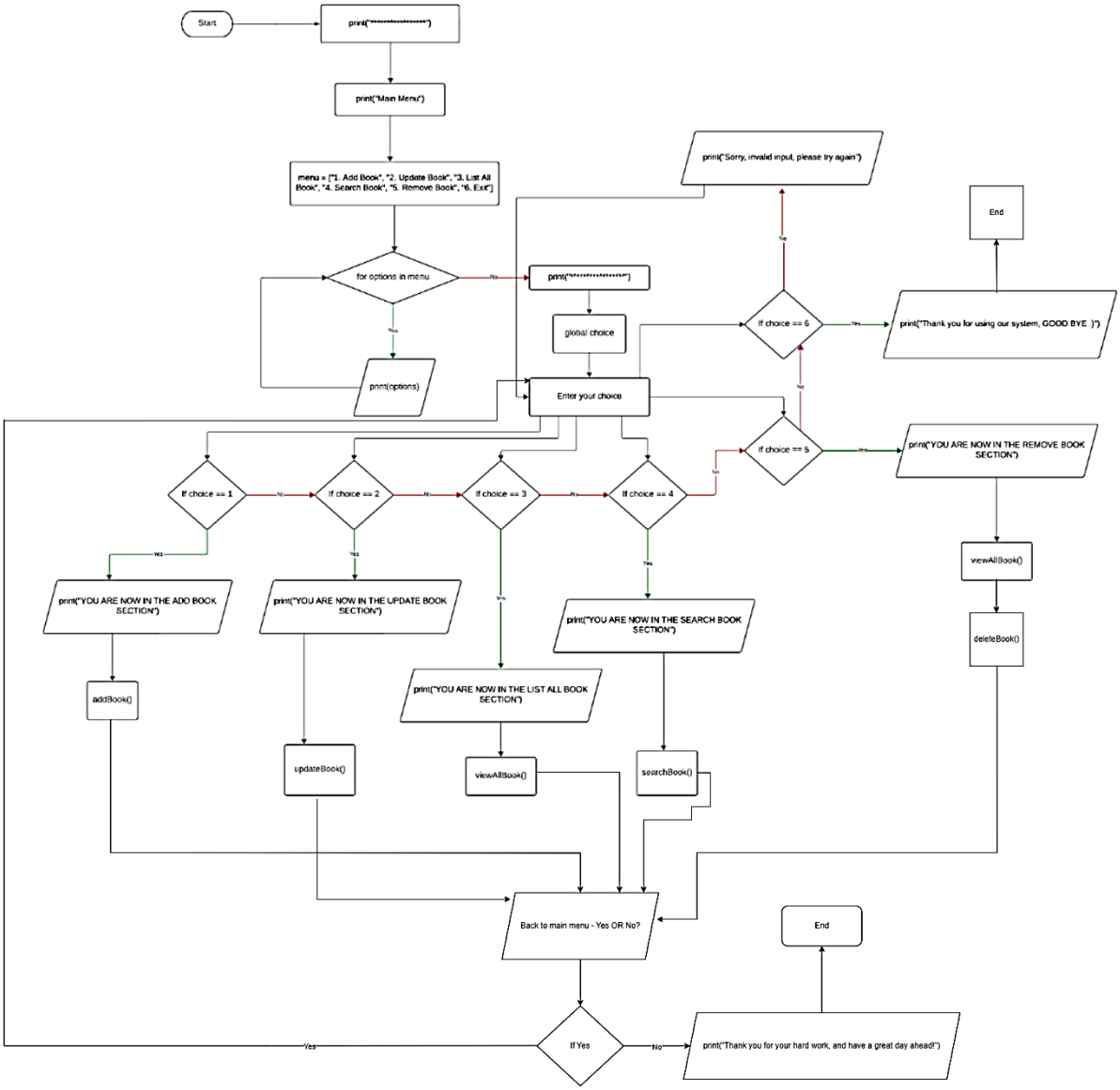
The library management system was developed to streamline the process for students and staff at Asia Pacific University (APU) so that they can easily access the system to add, update, remove and search for any particular book record as well as view the whole book list. Besides, this system can also help streamline the process of managing and organizing large volumes of books. Python is used as the foundation for this system as it is layman-friendly and easy to use to create any system.

There are several assumptions to be made when creating and examining the functionality of this system, as listed below:

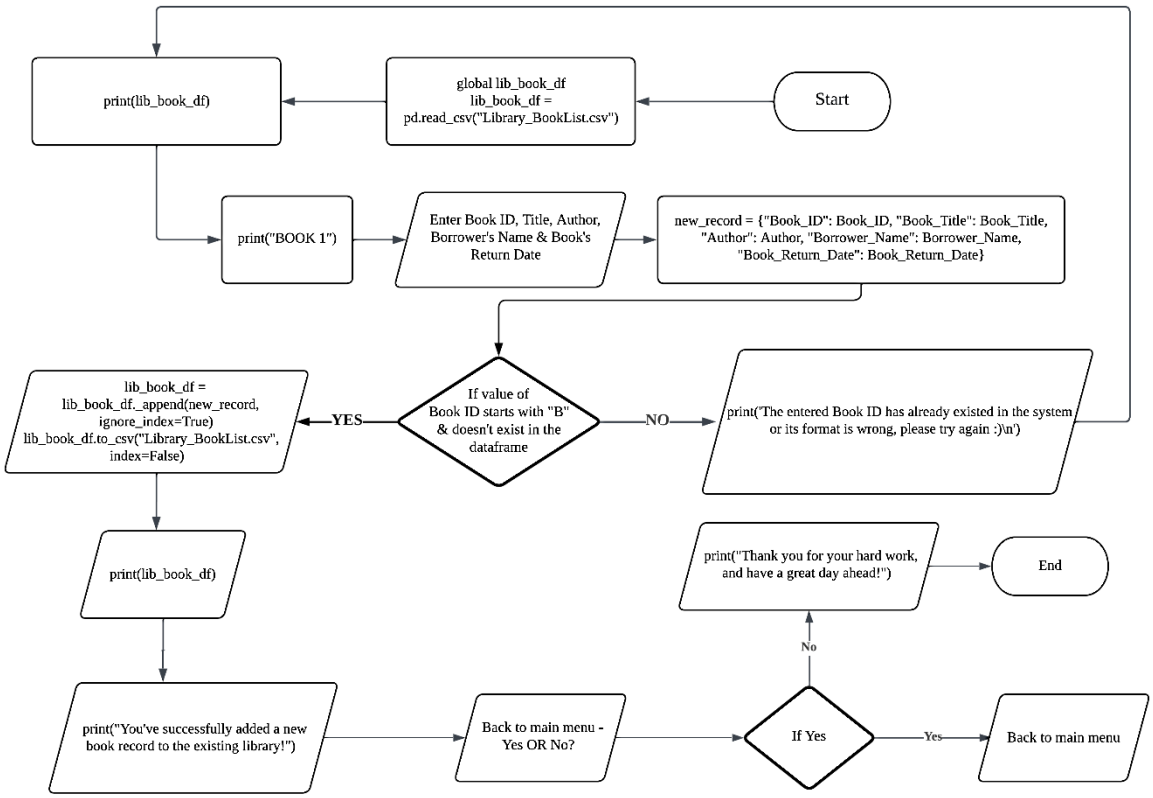
- The first assumption will revolve around file accessibility as we assume that Python has permission to read and manipulate the CSV file, named “Library_Booklist.csv”, that stores every book record.
- The second assumption is about the predetermined data format in the data frame and its corresponding CSV file, where there are five columns namely Book_ID, Book_Title, Author, Borrower_Name and Book_Return_Date, and each row in both sources corresponds to every book record.
- The third assumption is that users are required to input information according to the predetermined rule to avoid any form of validation and type errors. For example, users are restricted and required to input the Book_ID to validate the format of the ID number and its existence in the system before proceeding with the specific operation.
- The fourth assumption is that users can only append, update, search and delete one specific book record at a time.

Program Design (Flowchart)

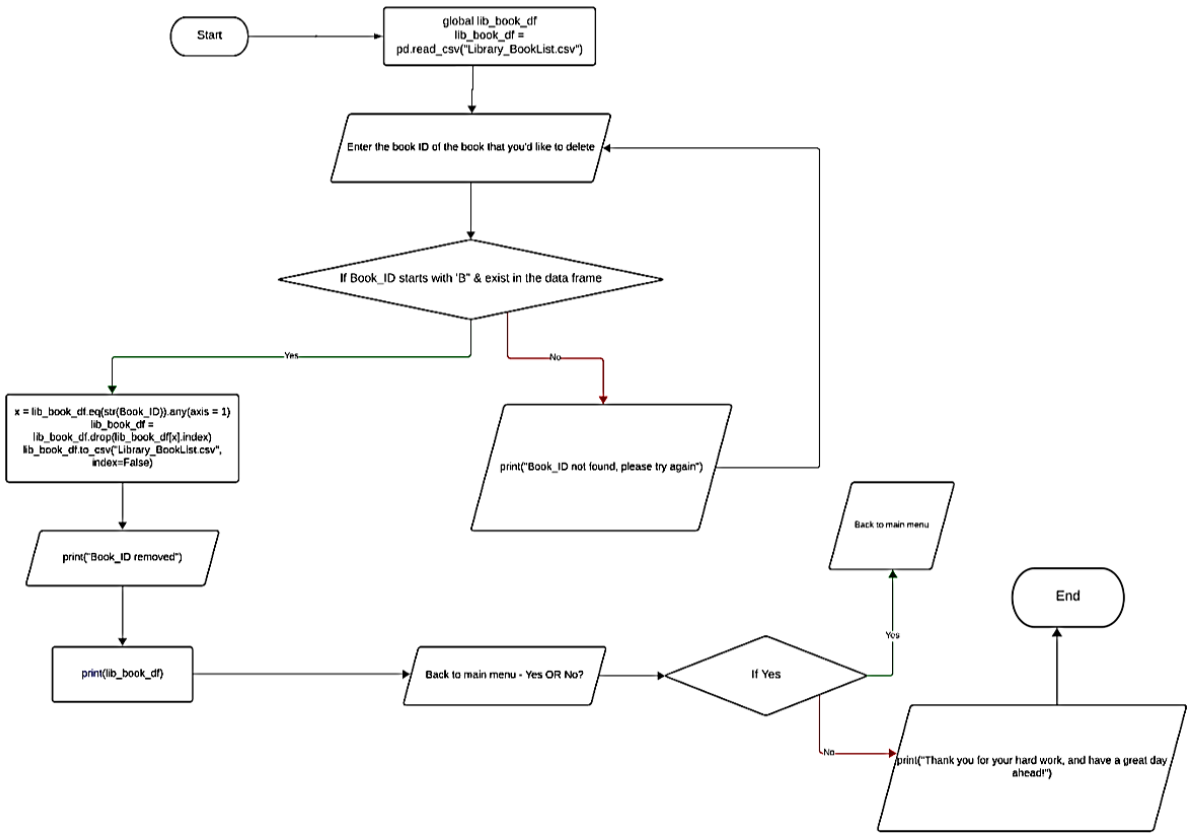
Program as a Whole



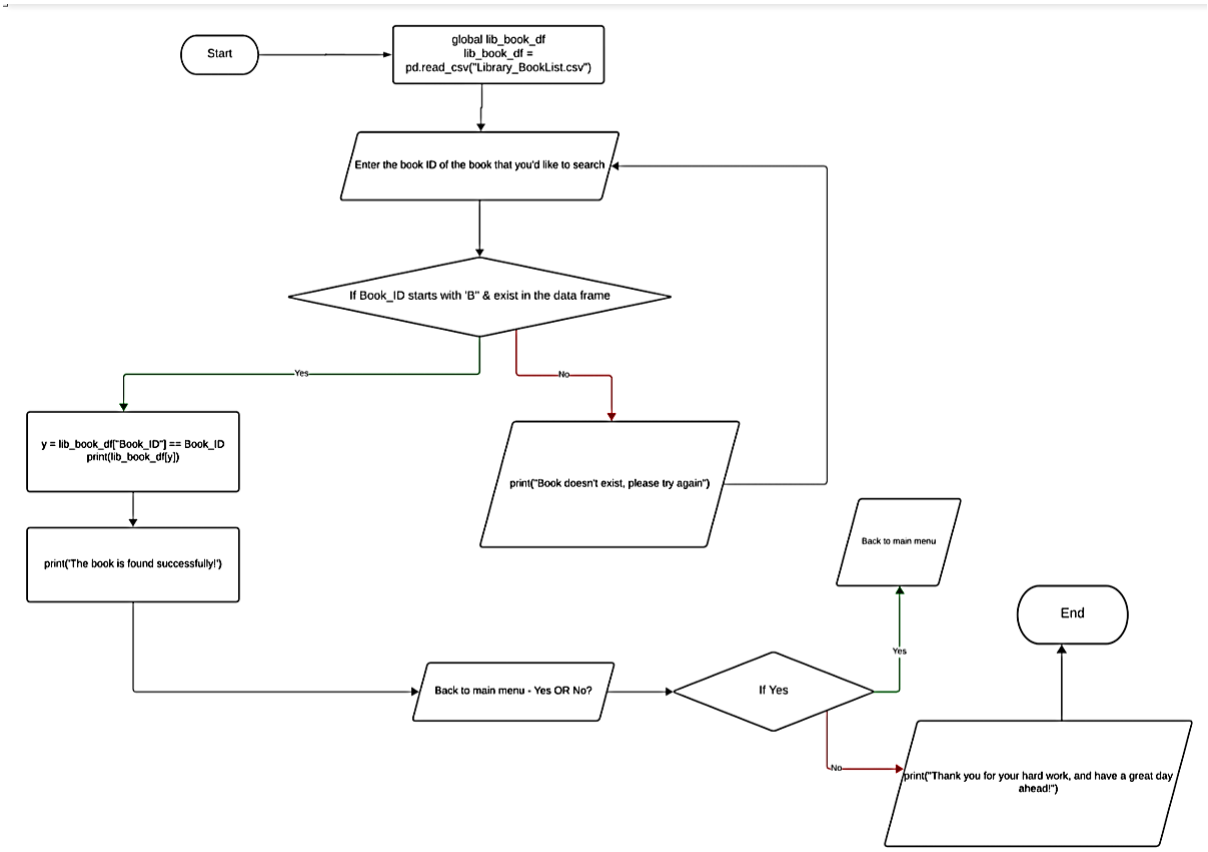
addBook()


print(lib_book_df)
global lib_book_df
lib_book_df =
pd.read_csv("Library_BookList.csv")
Start
print("BOOK 1")
Enter Book ID, Title, Author,
Borrower's Name & Book's
Return Date
new_record = {"Book_ID": Book_ID, "Book_Title": Book_Title,
"Author": Author, "Borrower_Name": Borrower_Name,
"Book_Return_Date": Book_Return_Date}
lib_book_df =
lib_book_df._append(new_record,
ignore_index=True)
lib_book_df.to_csv("Library_BookList.csv",
index=False)
YES
If value of
Book ID starts with "B"
& doesn't exist in the
dataframe
NO
print('The entered Book ID has already existed in the system
or its format is wrong, please try again :)\n')
print(lib_book_df)
print("Thank you for your hard work,
and have a great day ahead!")
End
No
print("You've successfully added a new
book record to the existing library!")
Back to main menu -
Yes OR No?
If Yes
Yes
Back to main menu

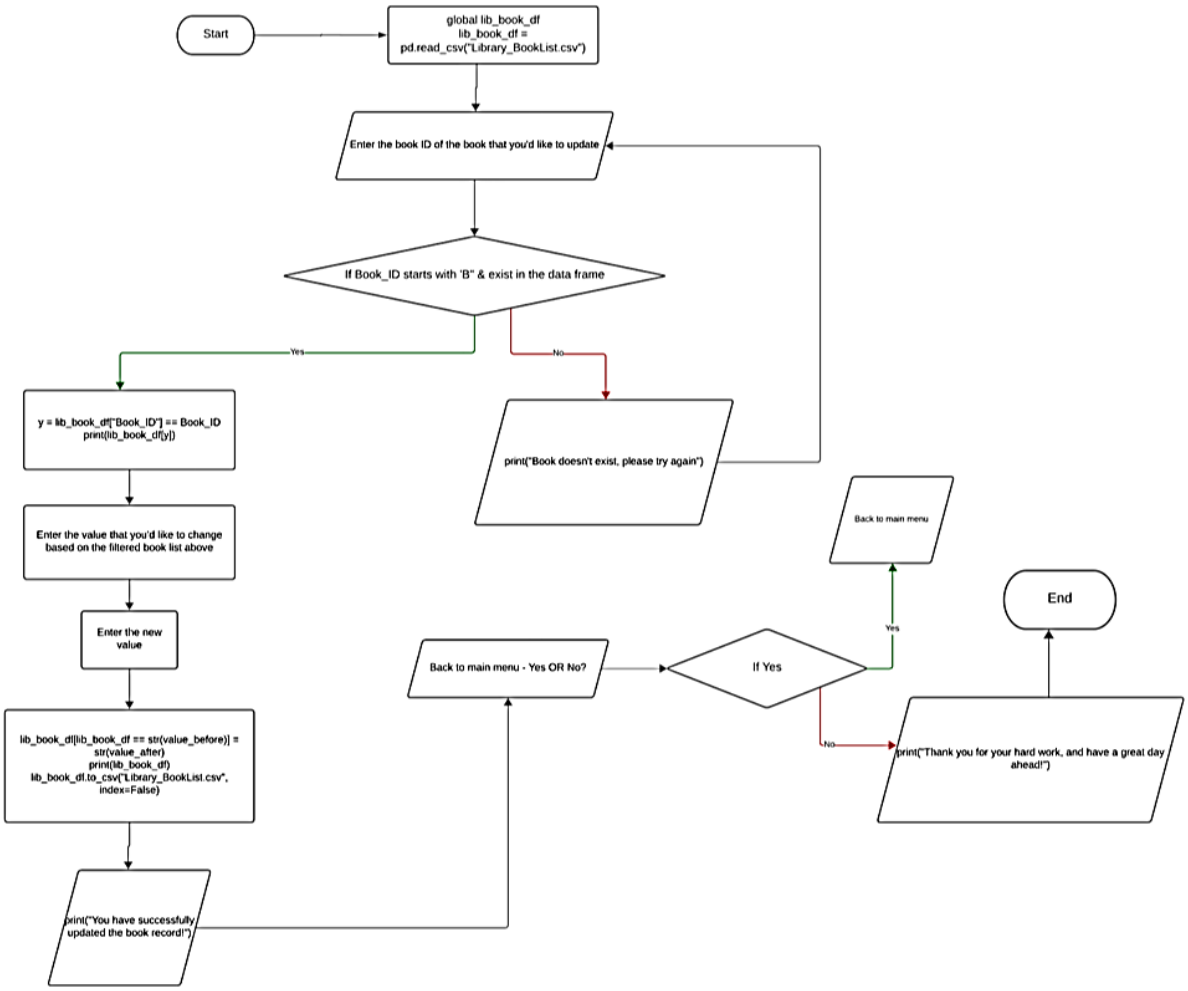
deleteBook()


Start
global lib_book_df
lib_book_df =
pd.read_csv("Library_BookList.csv")
Enter the book ID of the book that you'd like to delete
If Book_ID starts with 'B" & exist in the data frame
Yes
No
x = lib_book_df.eq(str(Book_ID)).any(axis = 1)
lib_book_df =
lib_book_df.drop(lib_book_df[x].index)
lib_book_df.to_csv("Library_BookList.csv",
index=False)
print("Book_ID not found, please try again")
print("Book_ID removed")
print(lib_book_df)
Back to main menu - Yes OR No?
If Yes
Yes
Back to main menu
End
No
print("Thank you for your hard work, and have a great day
ahead!")

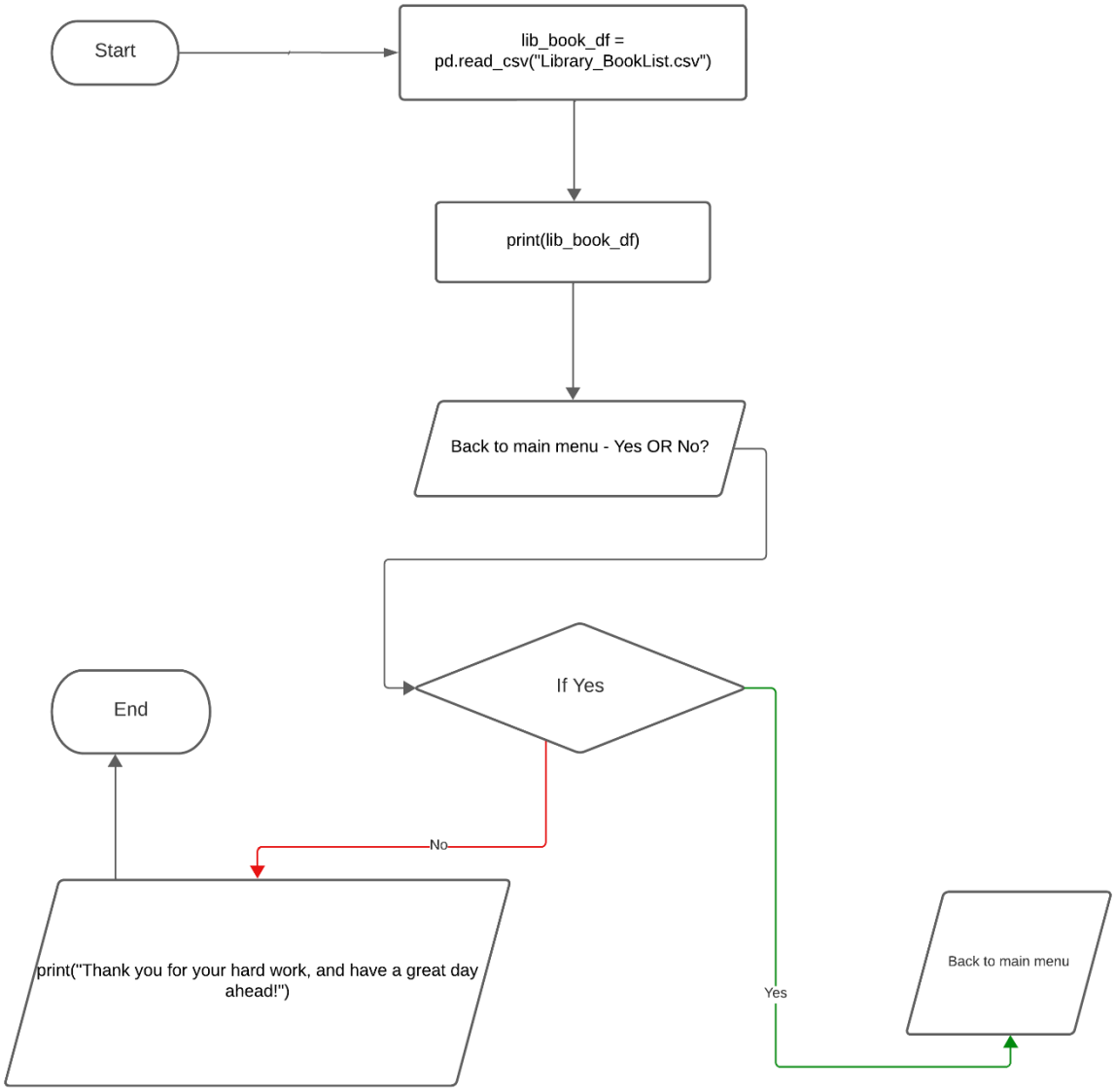
searchBook()


Start
global lib_book_df
lib_book_df =
pd.read_csv("Library_BookList.csv")
Enter the book ID of the book that you'd like to search
If Book_ID starts with 'B" & exist in the data frame
Yes
No
y = lib_book_df["Book_ID"] == Book_ID
print(lib_book_df[y])
print("Book doesn't exist, please try again")
print('The book is found successfully!')
Back to main menu
End
Back to main menu - Yes OR No?
If Yes
Yes
No
print("Thank you for your hard work, and have a great day ahead!")

updateBook()


Start
global lib_book_df
lib_book_df =
pd.read_csv("Library_BookList.csv")
Enter the book ID of the book that you'd like to update
If Book_ID starts with 'B" & exist in the data frame
Yes
No
y = lib_book_df["Book_ID"] == Book_ID
print(lib_book_df[y])
print("Book doesn't exist, please try again")
Enter the value that you'd like to change based on the filtered book list above
Enter the new value
lib_book_df[lib_book_df == str(value_before)] = str(value_after)
print(lib_book_df)
lib_book_df.to_csv("Library_BookList.csv", index=False)
print("You have successfully updated the book record!")
Back to main menu - Yes OR No?
If Yes
Yes
No
Back to main menu
End
print("Thank you for your hard work, and have a great day ahead!")

viewAllBook()



Program Design (Pseudocode)

Pseudocode - Notepad

File Edit Format View Help

```
Goal: To ensure users are able to choose the specific options from the main menu and proceed with performing the specific operations to the
      book records

# Create a sample dataframe and csv file for the library management system
STEP 1: A dictionary called "library" is created which stores 2 sample book records in key:list of values pairs as shown below:

library = {
    "Book_ID": ["B059", "B057"],
    "Book_Title": ["Psychology", "Life"],
    "Author": ["George F.", "William T."],
    "Borrower_Name": ["Ryan", "Alex"],
    "Book_Return_Date": ["14/10/2023", "26/10/2023"]
}

STEP 2: This "library" dictionary is converted to its data frame form as "lib_book_df"
STEP 3: Save the latest "lib_book_df" data frame to csv file as "Library_BookList.csv" to store the data permanently into the disk and retrieve it
        for any usage later
```

Pseudocode - Notepad

File Edit Format View Help

```
# Define every functions below
FUNCTION addBook()
    Declare global variable "lib_book_df"
    Library_BookList.csv file is read from the directory as data frame and assigned to the variable "lib_book_df"
    Display the data frame "lib_book_df" as output for easier reference
    Blank line

    Print the statement - BOOK 1 as the heading
    Blank line
    Input - Please insert the book ID in the B___ format (e.g., B123)
    Input - Please insert the book title
    Input - Please insert the author's name
    Input - Please insert the borrower's name
    Input - Please insert the date when the book should be returned by the borrower (dd/mm/yyyy)
    Blank line

    The 5 newly inputted values previously will be substituted as values into the respective key:value pairs of the dictionary as new record as shown below

    new_record = {"Book_ID": Book_ID, "Book_Title": Book_Title, "Author": Author, "Borrower_Name": Borrower_Name,
                  "Book_Return_Date": Book_Return_Date}

    IF Book_ID starts with "B" and doesn't exist in the data frame THEN
        The new record will be added into the "lib_book_df" data frame as the most bottom row by default
        Save the latest "lib_book_df" data frame to csv file as "Library_BookList.csv"
        Display the data frame "lib_book_df" as output
        Blank line
        Print the statement - You've successfully added a new book record to the existing library!
    ELSE
        Print statement then leave a blank line - The entered Book ID has already existed in the system or its format is wrong, please try again :)
        Call the FUNCTION addBook() - Go back to inputting the 5 new values again for the new record to be added into the data frame & csv file

    Blank line
END FUNCTION
```

Pseudocode - Notepad

File Edit Format View Help

```
FUNCTION deleteBook()
    Declare global variable "lib_book_df"
    Library_BookList.csv file is read from the directory as data frame and assigned to the variable "lib_book_df"
    Input: Please enter the book ID of the book that you'd like to delete
    Blank line

    IF Book_ID starts with "B" and already in the data frame THEN
        The specific row of book record containing the inputted book ID will be deleted from the "lib_book_df" data frame
        Save the latest "lib_book_df" data frame to csv file as "Library_BookList.csv"
        Print the statement - Book_ID removed
        Blank line
        Display the data frame "lib_book_df" as output
        Blank line
    ELSE
        Print the statement - Book_ID not found, please try again
        Blank line
        Call the FUNCTION deleteBook() - - Go back to inputting book ID again to search for book record to be deleted later
END FUNCTION

FUNCTION searchBook()
    Declare global variable "lib_book_df"
    Library_BookList.csv file is read from the directory as data frame and assigned to the variable "lib_book_df"
    Input: Please enter the book ID of the book that you'd like to search
    Blank line

    IF Book_ID starts with "B" and already in the data frame THEN
        Display the specific row of book record corresponding to the entered book ID as output
        Blank line
        Print the statement - The book is found successfully!
    ELSE
        Print the statement - Book doesn't exist, please try again
        Blank line
        Call the FUNCTION updateBook() - Go back to inputting book ID again to search for book record
END FUNCTION
```

Pseudocode - Notepad

File Edit Format View Help

```
FUNCTION updateBook()
    Declare global variable "lib_book_df"
    Library_BookList.csv file is read from the directory as data frame and assigned to the variable "lib_book_df"
    Input: Please enter the book ID of the book that you'd like to update
    Blank line

    IF Book_ID starts with "B" and already in the data frame THEN
        Display the specific row of book record corresponding to the entered book ID as output
        Blank line
        Input: Existing value that you'd like to change based on the specific row of book record
        Input: New value to update the specific row of book record
        Blank line
        Substitute the existing value with the new value
        Display the data frame "lib_book_df" as output
        Save the latest "lib_book_df" data frame to csv file as "Library_BookList.csv"
        Blank line
        Print the statement - You have successfully updated the book record!
    ELSE
        Print the statement - Book doesn't exist, please try again
        Blank line
        Call the FUNCTION updateBook() - Go back to inputting book ID again to update book record
END FUNCTION

FUNCTION viewAllBook()
    Library_BookList.csv file is read from the directory as data frame and assigned to the variable "lib_book_df"
    Display the data frame "lib_book_df" as output
END FUNCTION

FUNCTION main_menu()
    Print a line of **********
    Print the title called "Main Menu"
    Different options in the list (eg. Add Book, Update Book, List All Book, Remove Book, Search Book, Exit) are assigned to the variable "menu"
    FOR each options in the menu
        Iterate through every options and print each of them in seperate lines
    END FOR
    Print another line of **********
    Declare a global variable "choice"
    Input: Any specific choice among the options in the menu
    Blank line
END FUNCTION
```

Pseudocode - Notepad

File Edit Format View Help

```
# Combining the functions created above and integrate all of them into the overall main menu as shown below
Call the FUNCTION main_menu() and substitute it here

WHILE True
    IF your choice IS "1"
        Print the statement - YOU ARE NOW IN THE ADD BOOK SECTION
        Blank line
        Call the FUNCTION addBook() to add new book record
        Input: Back to main menu - Yes OR No?
        IF "Yes" THEN
            Blank line
            Call the FUNCTION main_menu() to return to main menu
            Blank line
        ELSE
            Blank line
            Print the statement - Thank you for your hard work, and have a great day ahead!
            Exit the program
    ELSE IF your choice IS "2"
        Print the statement - YOU ARE NOW IN THE UPDATE BOOK SECTION
        Blank line
        Call the FUNCTION updateBook() to update book record
        Blank line
        Input: Back to main menu - Yes OR No?
        IF "Yes" THEN
            Blank line
            Call the FUNCTION main_menu() to return to main menu
            Blank line
        ELSE
            Blank line
```

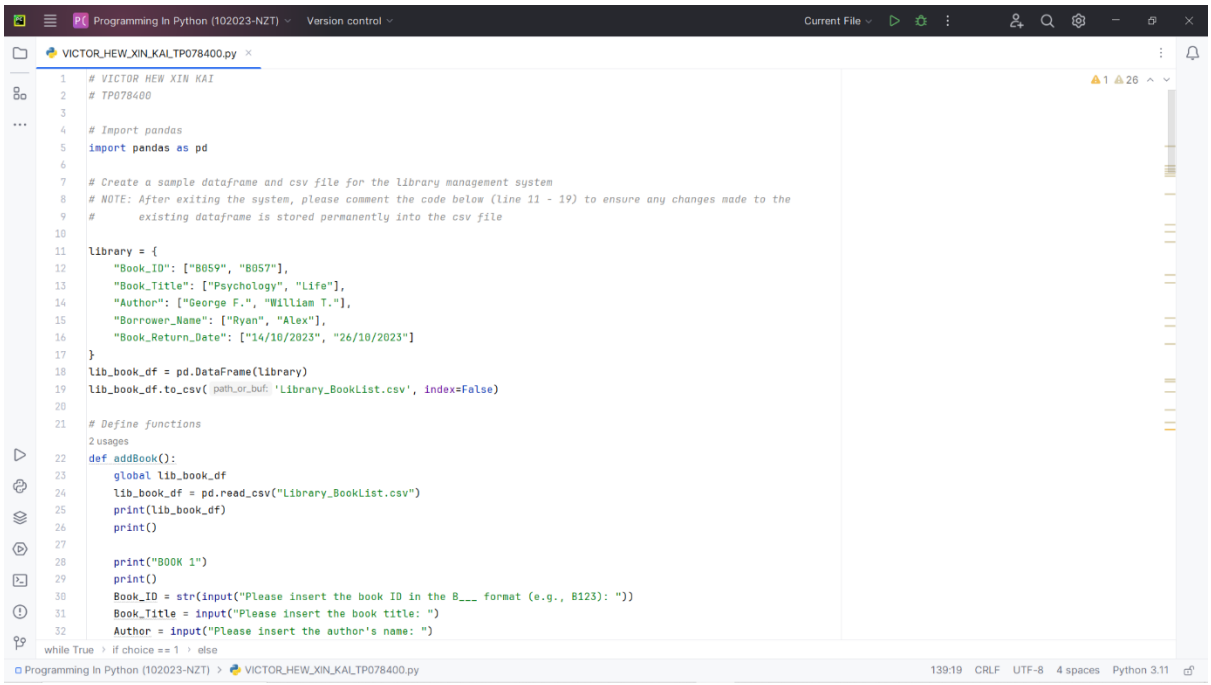
Pseudocode - Notepad

File Edit Format View Help

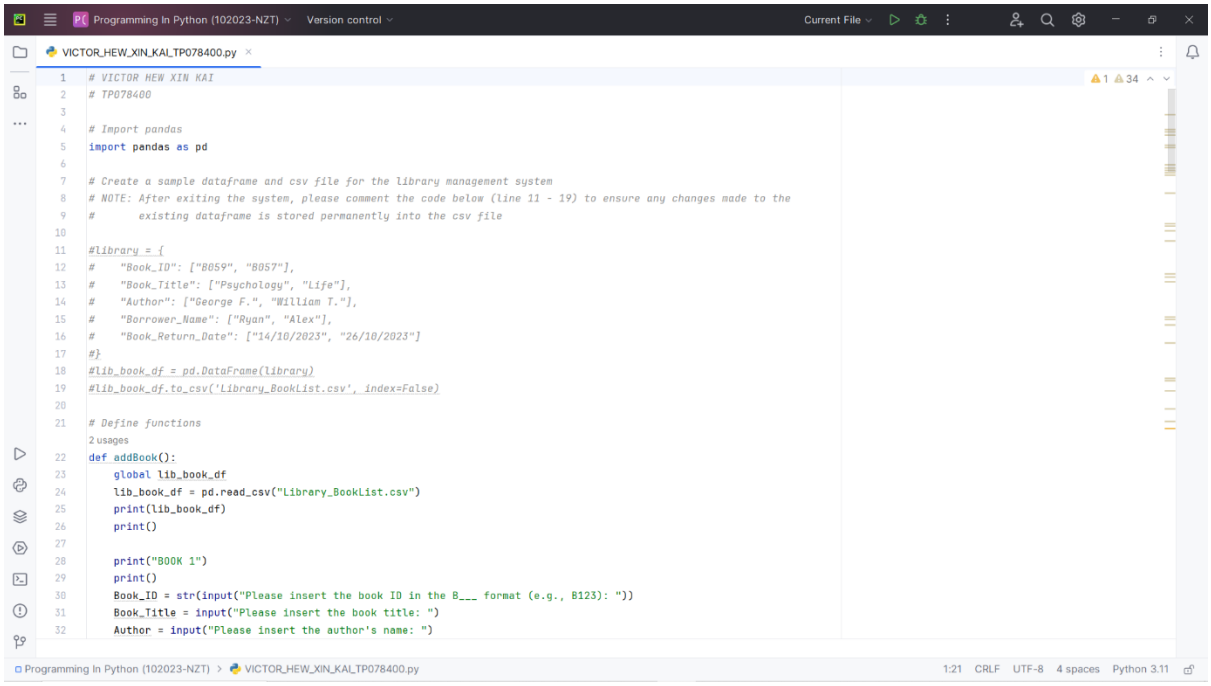
```
    ELSE IF your choice IS "3"
        Print the statement - YOU ARE NOW IN THE LIST ALL BOOK SECTION
        Blank line
        Call the FUNCTION viewAllBook() to list all book records
        Blank line
        Input: Back to main menu - Yes OR No?
        IF "Yes" THEN
            Blank line
            Call the FUNCTION main_menu() to return to main menu
            Blank line
        ELSE
            Blank line
            Print the statement - Thank you for your hard work, and have a great day ahead!
            Exit the program
    ELSE IF your choice IS "4"
        Print the statement - YOU ARE NOW IN THE SEARCH BOOK SECTION
        Blank line
        Call the FUNCTION searchBook() to search for any particular book record
        Blank line
        Input: Back to main menu - Yes OR No?
        IF "Yes" THEN
            Blank line
            Call the FUNCTION main_menu() to return to main menu
            Blank line
        ELSE
            Blank line
            Print the statement - Thank you for your hard work, and have a great day ahead!
            Exit the program
    ELSE IF your choice IS "5"
        Print the statement - YOU ARE NOW IN THE REMOVE BOOK SECTION
        Blank line
        Call the FUNCTION viewAllBook() to list all book records
        Blank line
        Call the FUNCTION deleteBook() to delete book record
        Input: Back to main menu - Yes OR No?
        IF "Yes" THEN
            Blank line
            Call the FUNCTION main_menu() to return to main menu
            Blank line
        ELSE
            Blank line
            Print the statement - Thank you for your hard work, and have a great day ahead!
            Exit the program



    ELSE IF your choice IS "6"
        Print the statement - Thank you for using our system, GOOD BYE :)
        Exit the program
    ELSE
        Print the statement - Sorry, invalid input, please try again"
        Blank line
        Call the FUNCTION main_menu() to return to the main menu page
```

Ln 129, Col 1 | 100% | Windows (CRLF) | UTF-8

Screenshots of Source Code



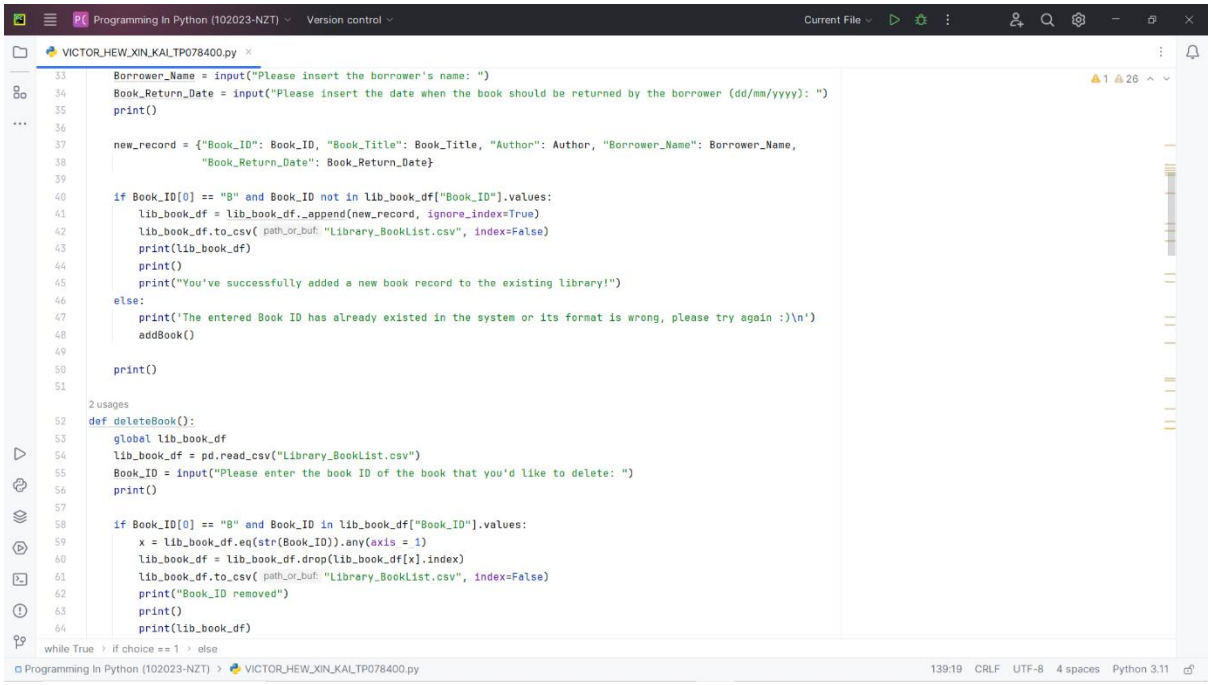
```
# VICTOR HEW XIN KAI
# TP078400

# Import pandas
import pandas as pd

# Create a sample dataframe and csv file for the library management system
# NOTE: After exiting the system, please comment the code below (line 11 - 19) to ensure any changes made to the
#       existing dataframe is stored permanently into the csv file

library = {
    "Book_ID": ["B059", "B057"],
    "Book_Title": ["Psychology", "Life"],
    "Author": ["George F.", "William T."],
    "Borrower_Name": ["Ryan", "Alex"],
    "Book_Return_Date": ["14/10/2023", "26/10/2023"]
}
lib_book_df = pd.DataFrame(library)
lib_book_df.to_csv( path_or_buf: 'Library_BookList.csv', index=False)

# Define functions
2 usages
def addBook():
    global lib_book_df
    lib_book_df = pd.read_csv("Library_BookList.csv")
    print(lib_book_df)
    print()

    print("BOOK 1")
    print()
    Book_ID = str(input("Please insert the book ID in the B___ format (e.g., B123): "))
    Book_Title = input("Please insert the book title: ")
    Author = input("Please insert the author's name: ")
```



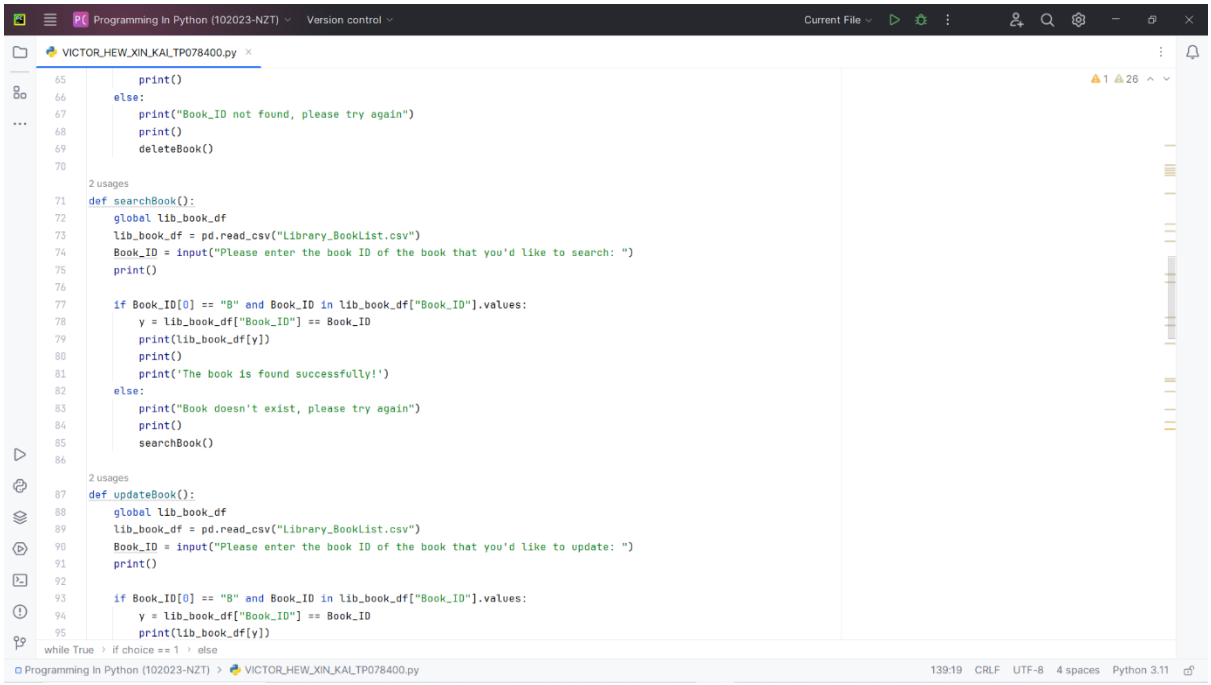
```
# VICTOR HEW XIN KAI
# TP078400

# Import pandas
import pandas as pd

# Create a sample dataframe and csv file for the library management system
# NOTE: After exiting the system, please comment the code below (line 11 - 19) to ensure any changes made to the
#       existing dataframe is stored permanently into the csv file

#library = {
#    "Book_ID": ["B059", "B057"],
#    "Book_Title": ["Psychology", "Life"],
#    "Author": ["George F.", "William T."],
#    "Borrower_Name": ["Ryan", "Alex"],
#    "Book_Return_Date": ["14/10/2023", "26/10/2023"]
#}
#lib_book_df = pd.DataFrame(library)
#lib_book_df.to_csv('Library_BookList.csv', index=False)

# Define functions
2 usages
def addBook():
    global lib_book_df
    lib_book_df = pd.read_csv("Library_BookList.csv")
    print(lib_book_df)
    print()

    print("BOOK 1")
    print()
    Book_ID = str(input("Please insert the book ID in the B___ format (e.g., B123): "))
    Book_Title = input("Please insert the book title: ")
    Author = input("Please insert the author's name: ")
```

Based on both screenshots above, after adding, deleting or updating any book record and then exiting the program for the very first attempt, the specific Python script for creating the sample book record needs to be commented on line-by-line. This is so that the changes made to the CSV file can be saved into the disk permanently instead of the memory, and that

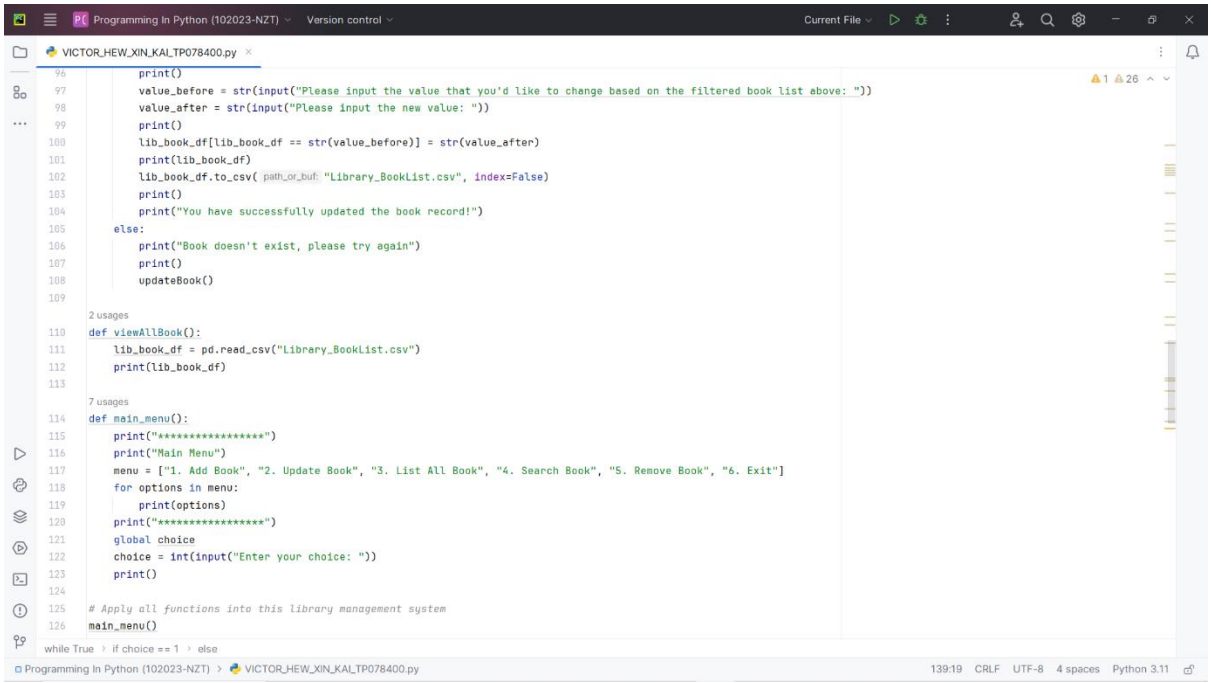
for subsequent attempts, if the same file is retrieved from the directory as data frame, the changes made previously will remain.



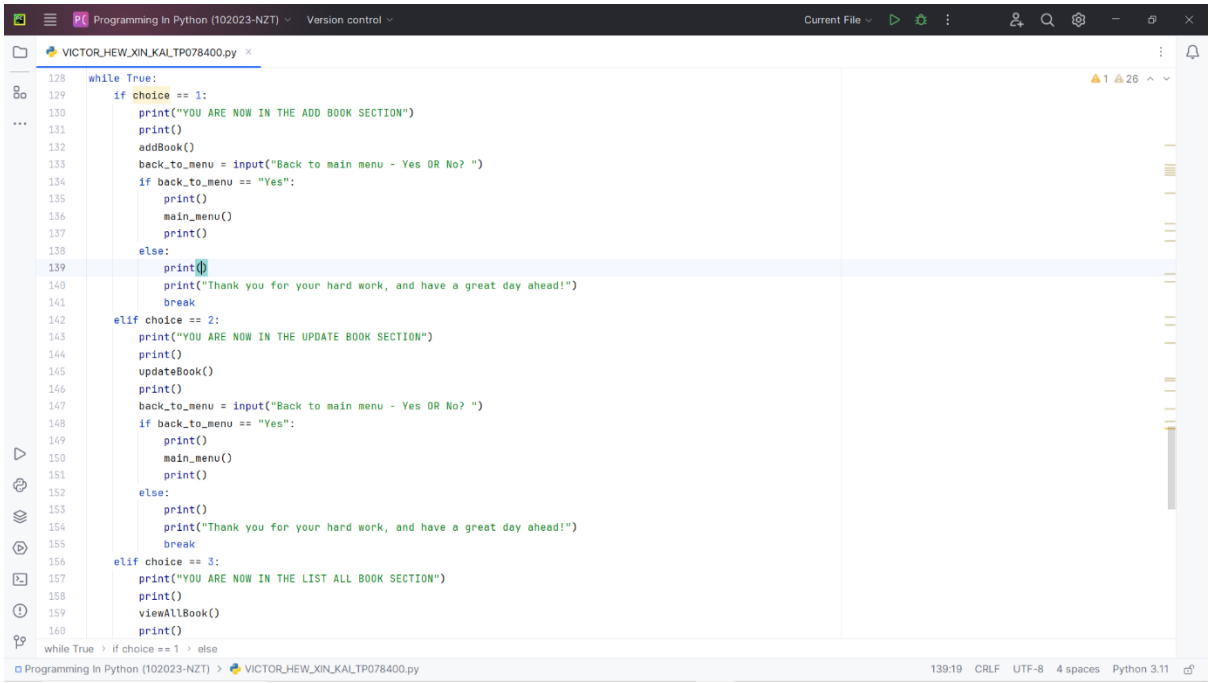
```
    Borrower_Name = input("Please insert the borrower's name: ")
    Book_Return_Date = input("Please insert the date when the book should be returned by the borrower (dd/mm/yyyy): ")
    print()

    new_record = {"Book_ID": Book_ID, "Book_Title": Book_Title, "Author": Author, "Borrower_Name": Borrower_Name,
                  "Book_Return_Date": Book_Return_Date}

    if Book_ID[0] == "B" and Book_ID not in lib_book_df["Book_ID"].values:
        lib_book_df = lib_book_df._append(new_record, ignore_index=True)
        lib_book_df.to_csv( path_or_buf: "Library_BookList.csv", index=False)
        print(lib_book_df)
        print()
        print("You've successfully added a new book record to the existing library!")
    else:
        print('The entered Book ID has already existed in the system or its format is wrong, please try again :)\n')
        addBook()

    print()

2 usages
def deleteBook():
    global lib_book_df
    lib_book_df = pd.read_csv("Library_BookList.csv")
    Book_ID = input("Please enter the book ID of the book that you'd like to delete: ")
    print()

    if Book_ID[0] == "B" and Book_ID in lib_book_df["Book_ID"].values:
        x = lib_book_df.eq(str(Book_ID)).any(axis = 1)
        lib_book_df = lib_book_df.drop(lib_book_df[x].index)
        lib_book_df.to_csv( path_or_buf: "Library_BookList.csv", index=False)
        print("Book_ID removed")
        print()
        print(lib_book_df)
```



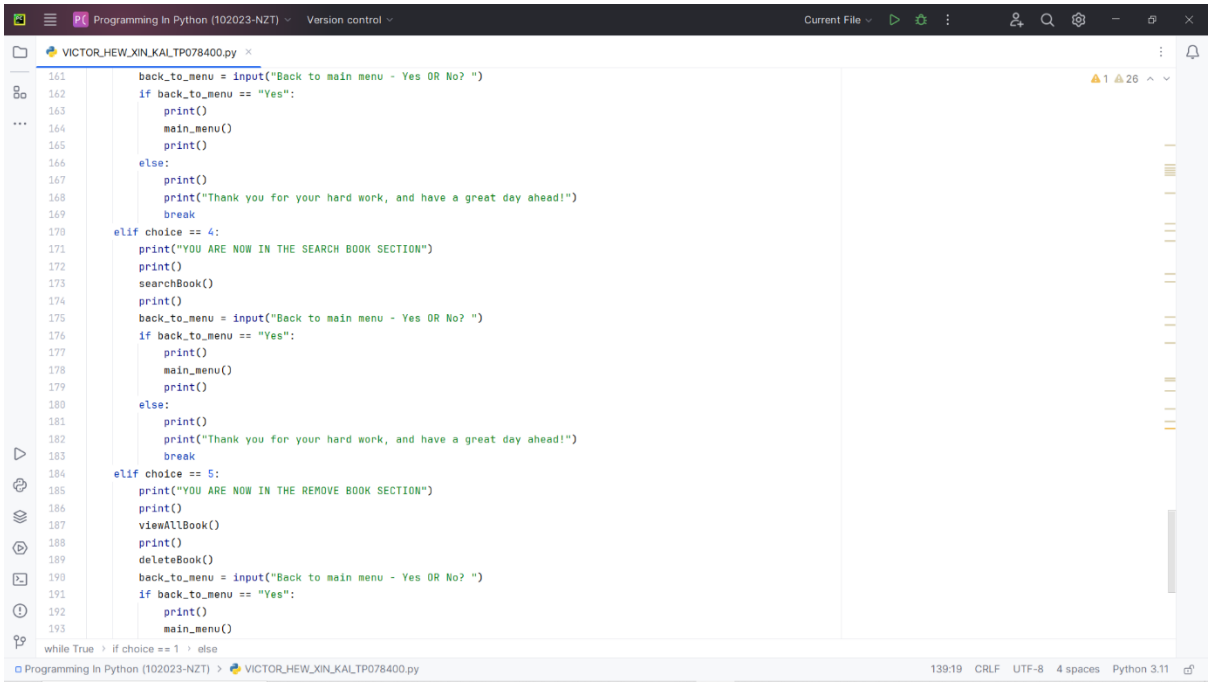
```
        print()
    else:
        print("Book_ID not found, please try again")
        print()
        deleteBook()

2 usages
def searchBook():
    global lib_book_df
    lib_book_df = pd.read_csv("Library_BookList.csv")
    Book_ID = input("Please enter the book ID of the book that you'd like to search: ")
    print()

    if Book_ID[0] == "B" and Book_ID in lib_book_df["Book_ID"].values:
        y = lib_book_df["Book_ID"] == Book_ID
        print(lib_book_df[y])
        print()
        print('The book is found successfully!')
    else:
        print("Book doesn't exist, please try again")
        print()
        searchBook()

2 usages
def updateBook():
    global lib_book_df
    lib_book_df = pd.read_csv("Library_BookList.csv")
    Book_ID = input("Please enter the book ID of the book that you'd like to update: ")
    print()

    if Book_ID[0] == "B" and Book_ID in lib_book_df["Book_ID"].values:
        y = lib_book_df["Book_ID"] == Book_ID
        print(lib_book_df[y])
```



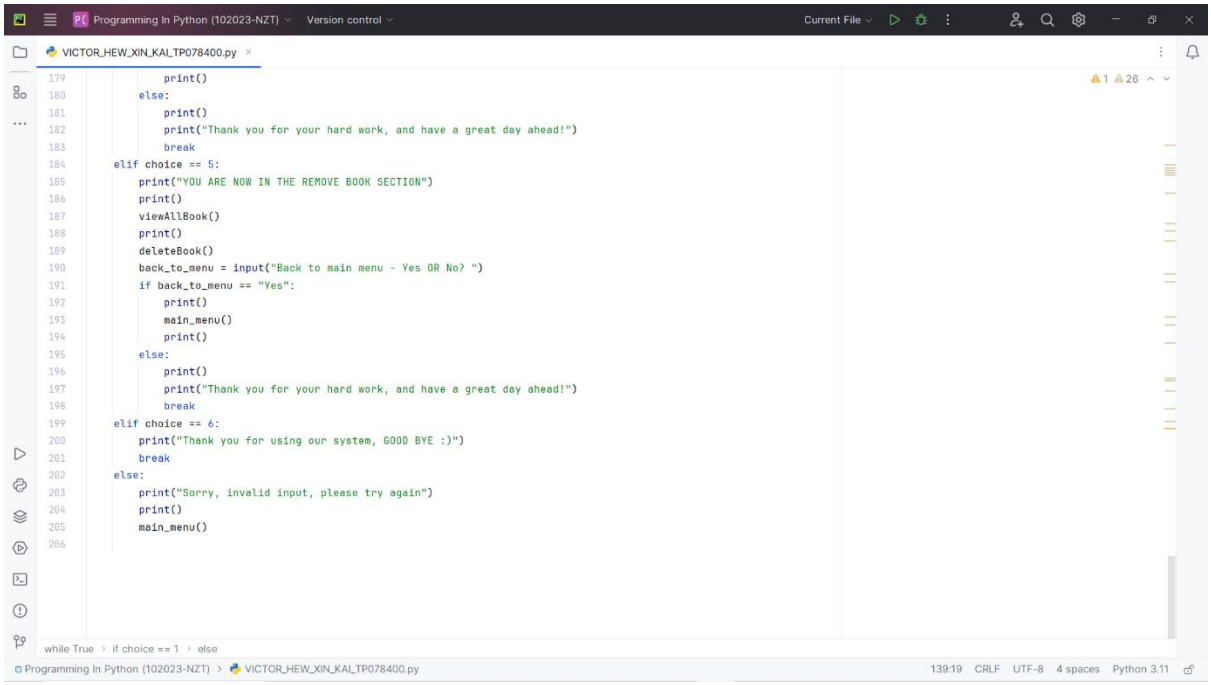
```
            print()
            value_before = str(input("Please input the value that you'd like to change based on the filtered book list above: "))
            value_after = str(input("Please input the new value: "))
            print()
            lib_book_df[lib_book_df == str(value_before)] = str(value_after)
            print(lib_book_df)
            lib_book_df.to_csv( path_or_buf: "Library_BookList.csv", index=False)
            print()
            print("You have successfully updated the book record!")
        else:
            print("Book doesn't exist, please try again")
            print()
            updateBook()

2 usages
def viewAllBook():
    lib_book_df = pd.read_csv("Library_BookList.csv")
    print(lib_book_df)

7 usages
def main_menu():
    print("******************")
    print("Main Menu")
    menu = ["1. Add Book", "2. Update Book", "3. List All Book", "4. Search Book", "5. Remove Book", "6. Exit"]
    for options in menu:
        print(options)
    print("******************")
    global choice
    choice = int(input("Enter your choice: "))
    print()

# Apply all functions into this library management system
main_menu()
```



```
while True:
    if choice == 1:
        print("YOU ARE NOW IN THE ADD BOOK SECTION")
        print()
        addBook()
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 2:
        print("YOU ARE NOW IN THE UPDATE BOOK SECTION")
        print()
        updateBook()
        print()
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 3:
        print("YOU ARE NOW IN THE LIST ALL BOOK SECTION")
        print()
        viewAllBook()
        print()
```



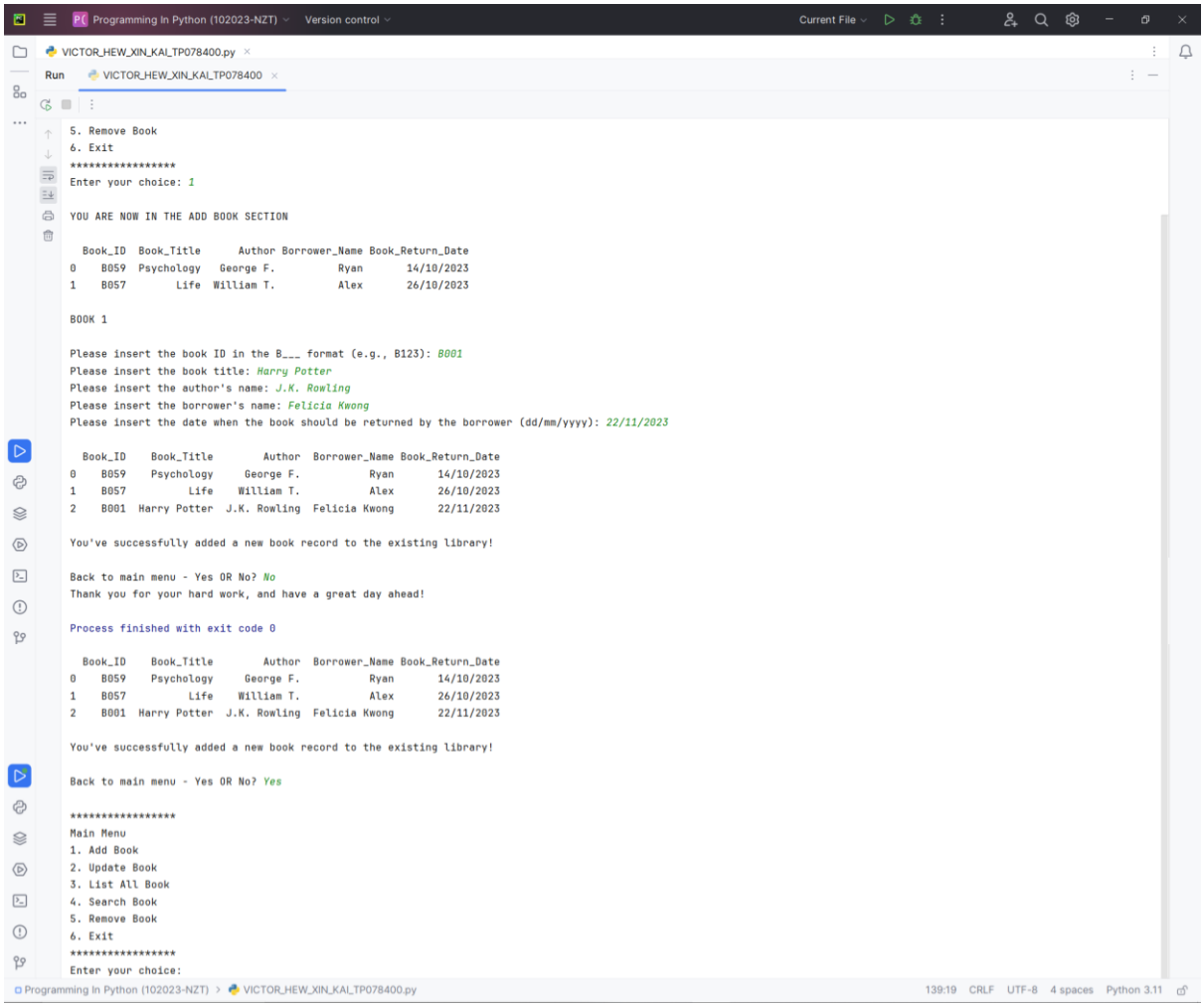
```
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 4:
        print("YOU ARE NOW IN THE SEARCH BOOK SECTION")
        print()
        searchBook()
        print()
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 5:
        print("YOU ARE NOW IN THE REMOVE BOOK SECTION")
        print()
        viewAllBook()
        print()
        deleteBook()
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
```



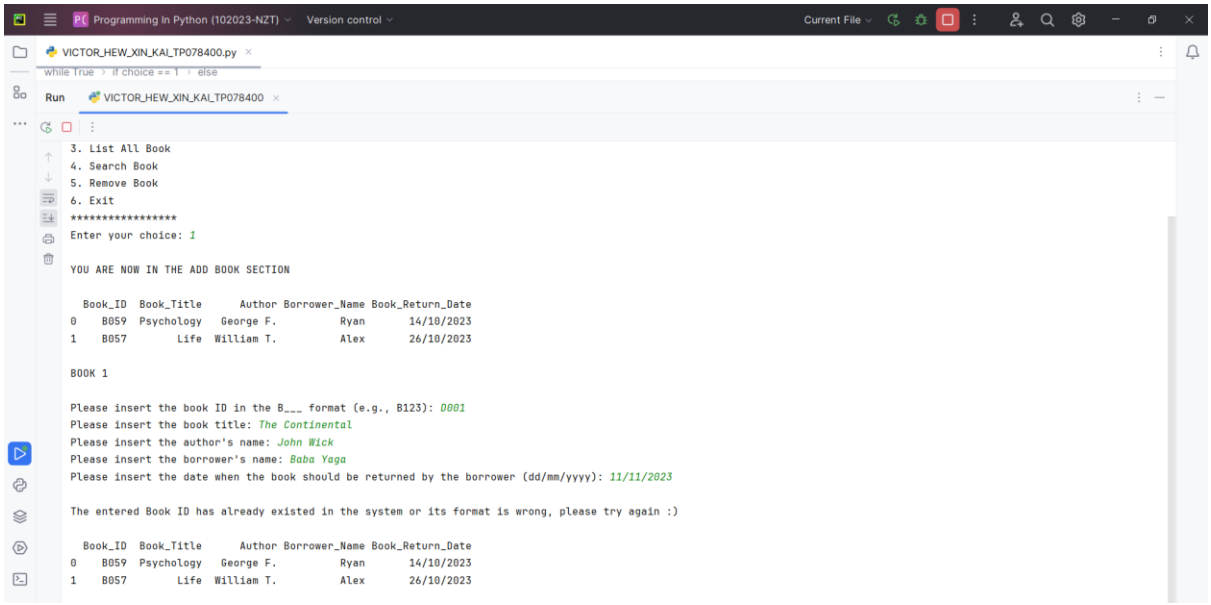
```
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 5:
        print("YOU ARE NOW IN THE REMOVE BOOK SECTION")
        print()
        viewAllBook()
        print()
        deleteBook()
        back_to_menu = input("Back to main menu - Yes OR No? ")
        if back_to_menu == "Yes":
            print()
            main_menu()
            print()
        else:
            print()
            print("Thank you for your hard work, and have a great day ahead!")
            break
    elif choice == 6:
        print("Thank you for using our system, GOOD BYE :)")
        break
    else:
        print("Sorry, invalid input, please try again")
        print()
        main_menu()
```

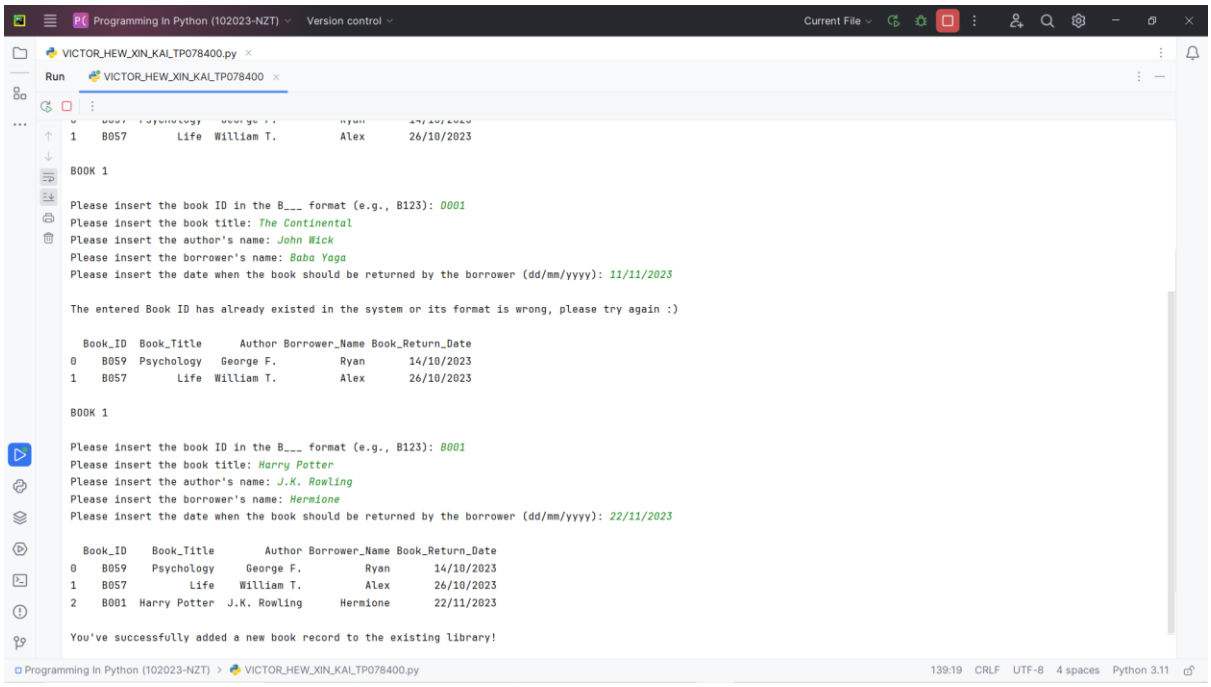
Screenshots of Sample Input/Output and Explanation

addBook()



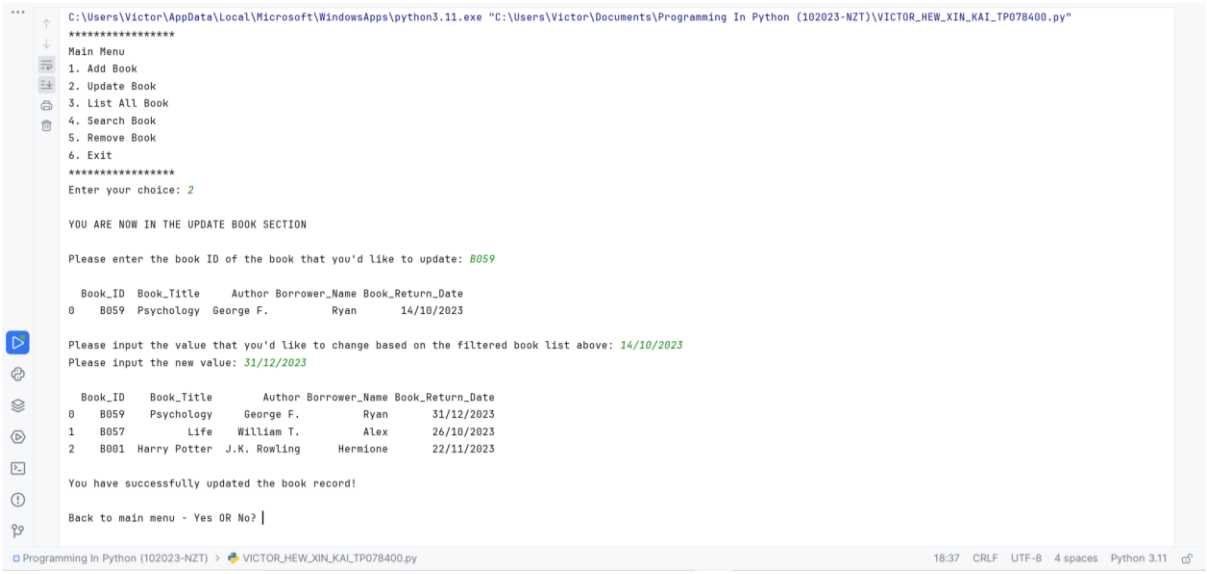
The main menu's first option, inputted as the value ‘1’, is to add one new book record to the existing data frame. As shown in the screenshots above, I have inputted the value “B001” as the book ID, “Harry Potter” as the book title, “J.K. Rowling” as the author, “Felicia Kwong” as the borrower’s name, and “22/11/2023” as the book’s return date. As a result, the new book record has been added to the existing data frame which is saved into the corresponding CSV file. In the end, I can either select “No” to exit the program or “Yes” to return to the main menu to proceed with other options.





In these two screenshots, I inputted the book ID in the wrong format as “D001” instead of IDs starting with “B”. Similarly, I can also input one of the existing book IDs, “B059” for the book ID, and both ways will cause the program to prompt me to try inserting another new book record again in the right way, which I succeeded in my second attempt using the same previous input except “Hermione” is inputted as author instead of “Felicia Kwong”. In the end, I can also either select “No” to exit the program or “Yes” to return to the main menu to proceed with other options.

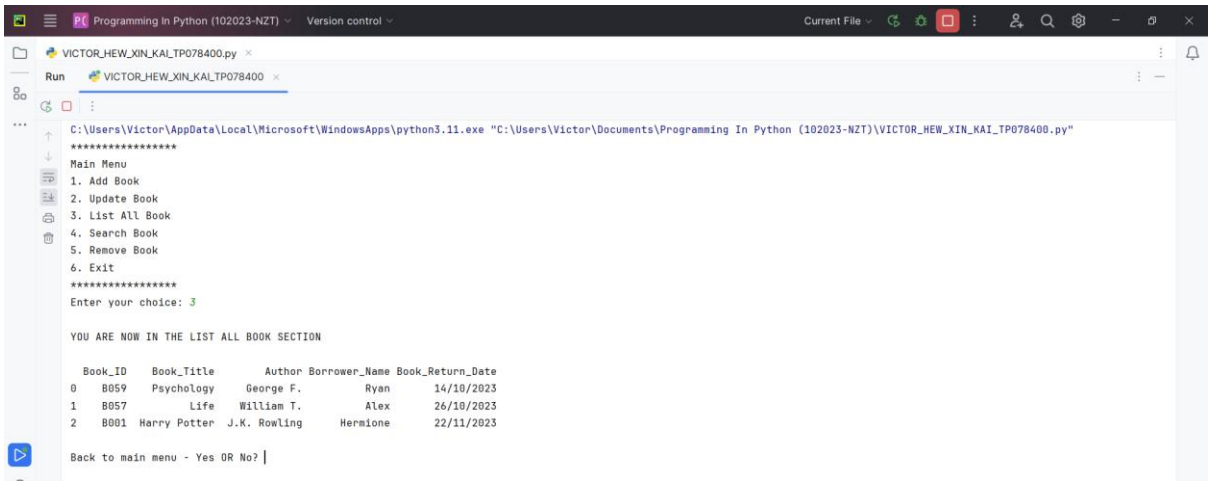
updateBook()



The main menu's second option, inputted as the value '2', is to update one existing book record in the data frame. As shown in the screenshot above, I have inputted the value "B059" as the book ID of the book record that I'd like to update. Then, I successfully updated the book return date of the book record from "14/10/2023" to "31/12/2023", which is saved into the corresponding CSV file. In the end, I can also either select "No" to exit the program or "Yes" to return to the main menu to proceed with other options.

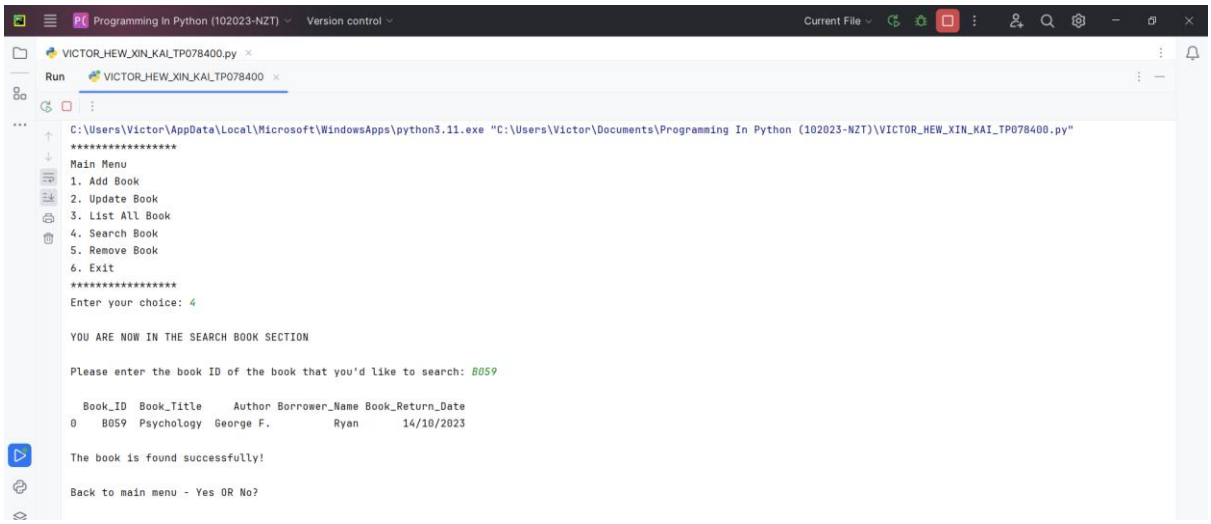
However, in the example here, I can also input a non-existing book ID like "D001" into the program and it shows that the book record doesn't exist, and I need to try again with an existing book ID like "B059" or "B057".

viewAllBook()

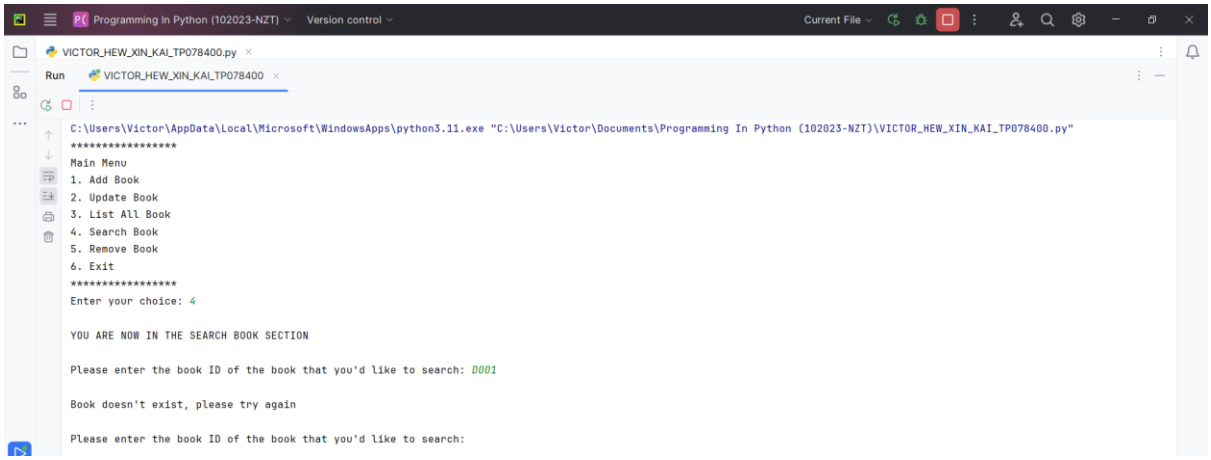


The main menu's third option, inputted as the value ‘3’, is to list all available book records in the data frame after any changes are made. In the end, I can also either select “No” to exit the program or “Yes” to return to the main menu to proceed with other options.

searchBook()

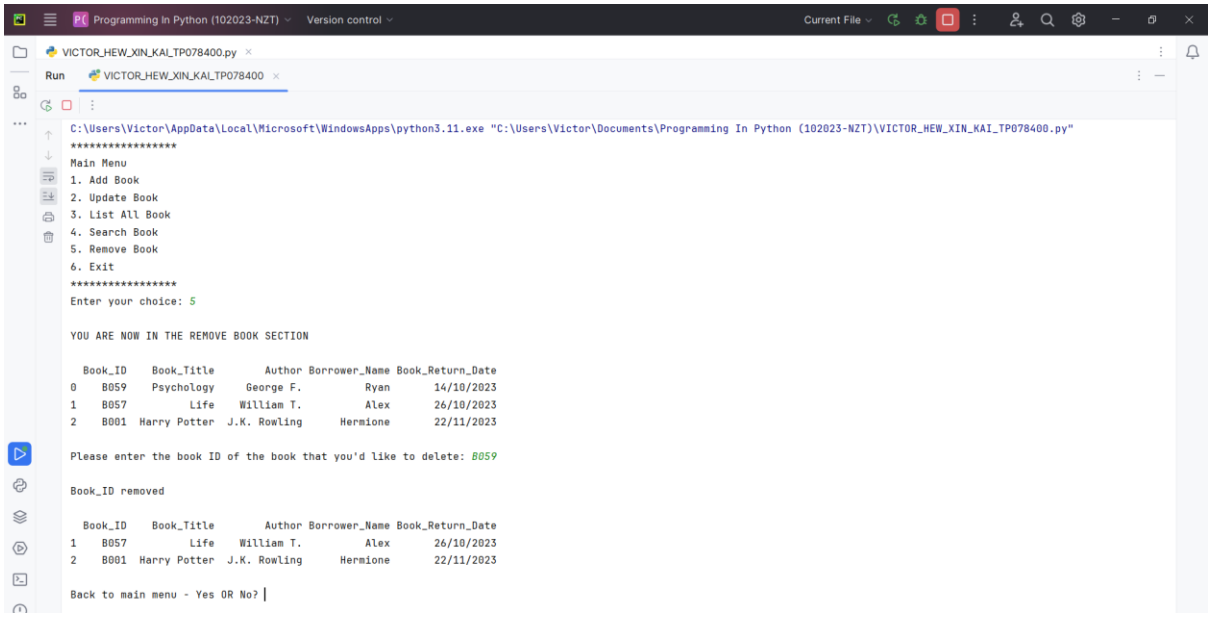


The main menu's fourth option, inputted as the value ‘4’, is to search for a particular book record in the existing data frame. Here, I’d like to search for a book record with the value “B059” being inputted as the book ID, and the corresponding book named “Psychology” is found eventually. In the end, I can also either select “No” to exit the program or “Yes” to return to the main menu to proceed with other options.

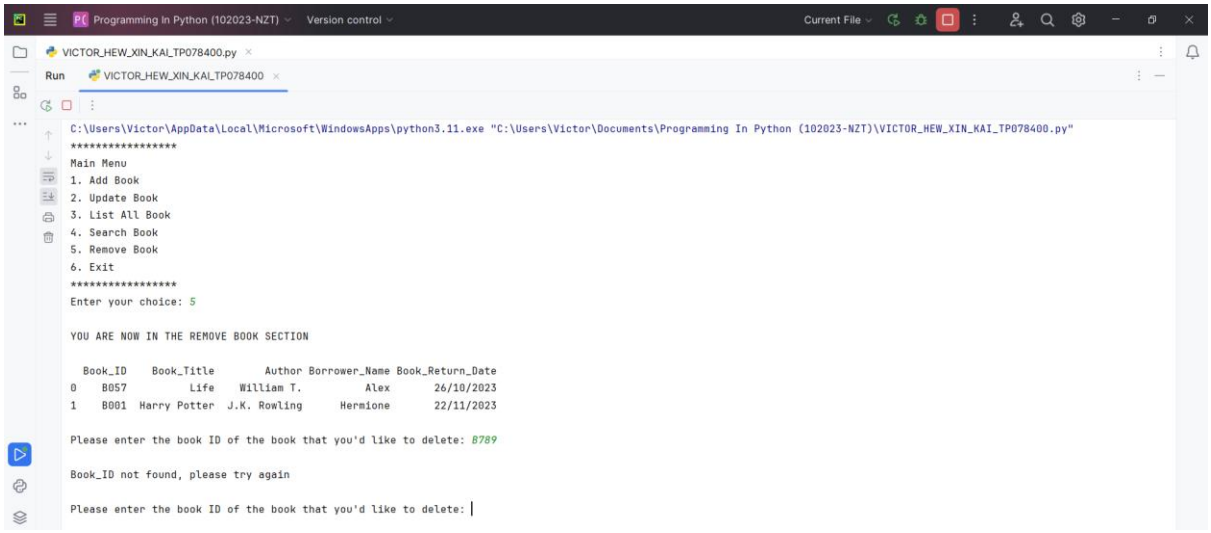


However, in the example here, I have inputted a non-existing book ID like “D001” into the program and it shows that the book record doesn’t exist, and I need to try searching again with an existing book ID like “B059” or “B057”.

deleteBook()

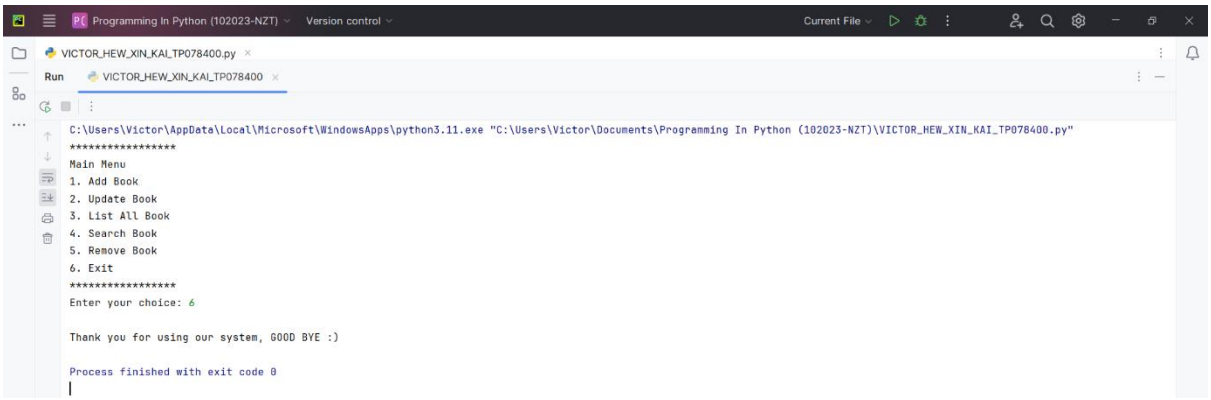


The main menu's fifth option, inputted as the value ‘5’, is to remove a particular book record from the existing data frame. Here, I’d like to delete the book record with the value “B059” being inputted as the book ID, and this record is deleted eventually. In the end, I can also either select “No” to exit the program or “Yes” to return to the main menu to proceed with other options.

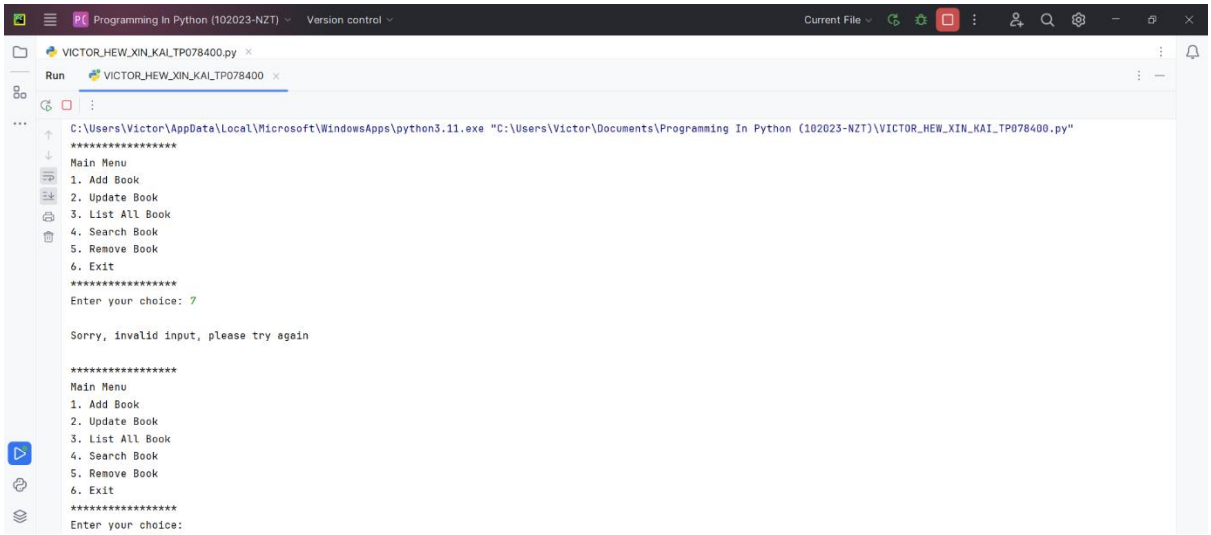


However, in the example here, I have inputted a non-existing book ID like “B789” into the program and it shows that the book record doesn’t exist, and I need to try again by inputting an existing book ID like “B057” in order to proceed with the delete operation.

Others



The main menu's fifth option, inputted as the value ‘6’, is to exit the program.



Let's say I inputted the value '7' as the option that I'd like to choose. The program sees it as an invalid input as it's not one of the options displayed in the main menu and I need to try again with the correct one, that are one of the six options displayed in the main menu.

Conclusion

All in all, this simple Python-based library management system can help streamline the book information tracking process. There are some suggestions for improvement so that the system can become all-around and more convenient for users. For example, adding other aspects to the data frame like the borrower's contact number and book lending date, as well as not limiting the program to add, delete or update book records one at a time.