**INSTITUTE OF SPACE TECHNOLOGY, ISLAMABAD**



**Lab Report – Project**

**Library Management System**

**Submitted To:**

Mr. Tufail Shah

**Submitted By:**

Abdullah Ashraf - 210201061

Abdul Basit – 210201023

Ahmed Zafar - 210201092

Muhammad Zaryab – 210201007

Muhammad Abdul Rehman – 210201088

**Fall-2021**

**Library Management System**

**Importing Modules**

import datetime

import pandas as pd

df = pd.read\_csv('Book1.csv')

today = datetime.date.today()

time\_to\_return = datetime.timedelta(days=14)  # 14 days to return

return\_time = today + time\_to\_return

structure = pd.DataFrame(df)

pd.set\_option('display.max\_columns', None)

pd.set\_option('display.width', 1000)

Here we are importing 2 modules “datetime” and “pandas”. Datetime is used for handling time usage and getting the current time from the user. Then we import a .csv file and store the current time in ‘today’ variable. The ‘time\_to\_return’ variable adds 14 days as a result date whenever it is called. ‘pd.set\_option’ is used to display the terminal correctly and prevent it from overlapping.

**Functions**

**Add Book**

def add\_book():

    global structure

    global today

    book = input("Enter new book: ")

    row = input("Enter book placement(row,column): ")

    newbook = {'Book Name': book, 'Entry Date': today, 'Availability': 'Yes', 'Issue Date': 'None', 'Issued to': 'None',

               'Return Date': 'None', 'Extra Charges': 0, 'Placement': row}

    structure = structure.append(newbook, ignore\_index=True)

The add\_book() function adds a new book in the DataFrame. Global variables are used so that any variable outside the function can be used inside the function as well. Then we take two inputs from the user for the book name and the placement of the book. Then using a Dictionary in which keys are column headers of a DataFrame, we append the entered data into the DataFrame.

**Issue Book**

def issue\_book():

    global structure

    global df

    print('-------------------------------------------------')

    print(structure)

    print('-------------------------------------------------')

    print('')

    book = input("What book do you want to issue?: ")

    if structure.loc[structure['Availability'] == 'No']:

        print("--")

    issue\_person = input("Who do you want to issue the book to: ")

    if structure['Book Name'].str.contains(book).any():

        structure.set\_index('Book Name', inplace=True)

        structure.loc[book, 'Issue Date'] = today

        structure.loc[book, 'Availability'] = 'No'

        structure.loc[book, 'Issued To'] = issue\_person

        structure.loc[book, 'Return Date'] = return\_time

        structure.reset\_index(inplace=True)

        print('-------------------------------------------------')

        print('Book issued Successfully')

        print('-------------------------------------------------')

    else:

        print('-------------------------------------------------')

        print("Wrong Input, Enter book index again!")

        print('-------------------------------------------------')

        issue\_book()

The issue\_book() function is used to issue a book to a person. It takes two inputs: one for the book name and the other for the person name it is issued to.

**Return Book**

def return\_book():

    global structure

    returned\_book = input("Enter book name to return: ")

    if structure['Book Name'].str.contains(returned\_book).any():

        structure.set\_index('Book Name', inplace=True)

        structure.loc[returned\_book, 'Availability'] = 'Yes'

        structure.loc[returned\_book, 'Issue Date'] = 'None'

        structure.loc[returned\_book, 'Return Date'] = 'None'

        structure.loc[returned\_book, 'Issued To'] = 'None'

        structure.reset\_index(inplace=True)

    else:

        print('-------------------------------------------------')

        print('Invalid Book')

        print('-------------------------------------------------')

        print('Enter Again')

        return\_book()

The return\_book() function returns an issued book and sets all the columns to default values.

**Available Function**

def available():

    global structure

    print(structure[structure['Availability'] == 'Yes'])

The available() function displays all books that are available to be issued.

**Delete Book**

def delete\_book():

    global structure

    delete = input("Which book do you want to delete?: ")

    if structure['Book Name'].str.contains(delete).any():

        structure.set\_index('Book Name', inplace=True)

        structure = structure.drop(delete)

        structure.reset\_index(inplace=True)

    else:

        print("Invalid Entry")

        print("Enter Again")

        delete\_book()

The delete\_book() function deletes a book from the DataFrame by deleting the row using drop function. Then we set the index to “Book Name” again so that we can take input from the user in a string.

**View Books Issued to a User**

def user\_issued():

    user = input("Enter user to view: ")

    global structure

    if structure['Issued To'].str.contains(user).any():  # Checks if the user is in the 'Issued To' box

        check = structure[structure['Issued To'] == user]

        print(check)

    else:

        print('No book issues to specifies person')

        print('')

        print('Enter again')

        user\_issued()

The user\_issued() function checks if the user is in the ‘Issued to’ column of the DataFrame and shows all the books issued to that user.

**Status of a Book**

def book\_issued():

    book = input("Enter Book to view: ")

    global structure

    if structure['Book Name'].str.contains(book).any():

        check = structure[structure['Book Name'] == book]

        print(check)

    else:

        print('-------------------------------------------------')

        print('No book with the name', book, 'found in the database')

        print('-------------------------------------------------')

        print('Enter Again')

        book\_issued()

The book\_issued() function checks if the entered book is in the DataFrame. If it is, then it shows the status of the book e.g. it’s availability, the user this book is issued to, placement of the book and entry date.

**Note:**

Else Statements at the end of each function is used to call the function again if an invalid entry is entered.

**Home Screen**

This the main code of the program where you are first welcomed by a welcome message and then using the while True infinite loop, we take inputs from user to choose. Each choice has its own function that it calls and if the choice entered is invalid, then the loop runs again.

**Welcome Screen**

print('-------------------------------------------------')

print('-                    -')

print('-  Welcome to Library Management System  -')

print('-                    -')

print('-------------------------------------------------')

print('')

As you can here, You are first welcomed by this message. Then we run a while True infinite loop.

**While True Loop**

while True:

    print("""           Enter 'A' to add a new book

           Enter 'I' to issue a book

           Enter 'U' to view issued books to the user

           Enter 'R' to return a book

           Enter 'V' to view available books

           Enter 'B' to view status of a book

           Enter 'D' to view database

           Enter 'X' to delete a book

           Enter 'Q' to exit

           Enter 'S' to save file""")

    print('-------------------------------------------------')

    choice = input("What would you like to do: ")

    print('-------------------------------------------------')

    if choice == 'A':

        add\_book()

        print(structure)

        print('-------------------------------------------------')

    elif choice == 'I':

        issue\_book()

        print('-------------------------------------------------')

    elif choice == 'U':

        user\_issued()

        print('-------------------------------------------------')

    elif choice == 'R':

        return\_book()

        print('-------------------------------------------------')

    elif choice == 'B':

        book\_issued()

        print('-------------------------------------------------')

    elif choice == 'D':

        print(structure)

        print('-------------------------------------------------')

    elif choice == 'X':

        delete\_book()

        print('-------------------------------------------------')

    elif choice == 'V':

        available()

        print('-------------------------------------------------')

Here what choice ‘D’ does is, it prints the whole DataFrame with all the books and information in it.

    elif choice == 'S':

        structure.to\_csv('Book1.csv', index=False)

        count = count + 1

    elif choice == 'Q':

        if count == 0:

            print("You have not saved your work. Do you want to exit anyways?:")

            decision = input(">")

            if decision == 'Yes':

                break

            else:

                continue

        else:

            break

    else:

        print('-------------------------------------------------')

        print('Invalid choice, Choose again')

        print('-------------------------------------------------')

print("Good Bye")

At the end, if the user wants to quit, the count variable is used to check if the user has saved the work or not. When user calls the save function, the count is added by 1 and the program breaks out of the loop and exits.

**Our Motive**

The reason behind this code is that we wanted to create a system where you can manage and add data in it and upon looking at the library management system, we found it to be very interesting. We wanted to design a program where you can have all your data in a single frame, save the new entered data and is also capable of performing all the other library management system related functions.

**Raw Code**

import datetime

import pandas as pd

df = pd.read\_csv('Book1.csv')

today = datetime.date.today()

time\_to\_return = datetime.timedelta(days=14) # 14 days to return

return\_time = today + time\_to\_return

structure = pd.DataFrame(df)

pd.set\_option('display.max\_columns', None)

pd.set\_option('display.width', 1000)

count = 0

def add\_book():

global structure

global today

book = input("Enter new book: ")

row = input("Enter book placement(row,column): ")

newbook = {'Book Name': book, 'Entry Date': today, 'Availability': 'Yes', 'Issue Date': 'None', 'Issued to': 'None',

'Return Date': 'None', 'Extra Charges': 0, 'Placement': row}

structure = structure.append(newbook, ignore\_index=True)

def issue\_book():

global structure

global df

print('------------------------------')

print(structure)

print('------------------------------')

print('')

book = input("What book do you want to issue?: ")

issue\_person = input("Who do you want to issue the book to: ")

if structure['Book Name'].str.contains(book).any():

structure.set\_index('Book Name', inplace=True)

structure.loc[book, 'Issue Date'] = today

structure.loc[book, 'Availability'] = 'No'

structure.loc[book, 'Issued To'] = issue\_person

structure.loc[book, 'Return Date'] = return\_time

structure.reset\_index(inplace=True)

print('------------------------------')

print('Book issued Successfully')

print('------------------------------')

else:

print('------------------------------')

print("Wrong Input, Enter book index again!")

print('------------------------------')

issue\_book()

def user\_issued():

user = input("Enter user to view: ")

global structure

if structure['Issued To'].str.contains(user).any(): # Checks if the user is in the 'Issued To' box

check = structure[structure['Issued To'] == user]

print(check)

else:

print('No book issues to specifies person')

print('')

print('Enter again')

user\_issued()

def book\_issued():

book = input("Enter Book to view: ")

global structure

if structure['Book Name'].str.contains(book).any():

check = structure[structure['Book Name'] == book]

print(check)

else:

print('------------------------------')

print('No book with the name', book, 'found in the database')

print('------------------------------')

print('Enter Again')

book\_issued()

def return\_book():

global structure

returned\_book = input("Enter book name to return: ")

if structure['Book Name'].str.contains(returned\_book).any():

structure.set\_index('Book Name', inplace=True)

structure.loc[returned\_book, 'Availability'] = 'Yes'

structure.loc[returned\_book, 'Issue Date'] = 'None'

structure.loc[returned\_book, 'Return Date'] = 'None'

structure.loc[returned\_book, 'Issued To'] = 'None'

structure.reset\_index(inplace=True)

else:

print('------------------------------')

print('Invalid Book')

print('------------------------------')

print('Enter Again')

return\_book()

def available():

global structure

print(structure[structure['Availability'] == 'Yes'])

def delete\_book():

global structure

delete = input("Which book do you want to delete?: ")

if structure['Book Name'].str.contains(delete).any():

structure.set\_index('Book Name', inplace=True)

structure = structure.drop(delete)

structure.reset\_index(inplace=True)

else:

print("Invalid Entry")

print("Enter Again")

delete\_book()

print('-------------------------------------------------')

print('- -')

print('- Welcome to Library Management System -')

print('- -')

print('-------------------------------------------------')

print('')

while True:

print(""" Enter 'A' to add a new book

Enter 'I' to issue a book

Enter 'U' to view issued books to the user

Enter 'R' to return a book

Enter 'V' to view available books

Enter 'B' to view status of a book

Enter 'D' to view database

Enter 'X' to delete a book

Enter 'Q' to exit

Enter 'S' to save file""")

print('-------------------------------------------------')

choice = input("What would you like to do: ")

print('-------------------------------------------------')

if choice == 'A':

add\_book()

print(structure)

print('-------------------------------------------------')

elif choice == 'I':

issue\_book()

print('-------------------------------------------------')

elif choice == 'U':

user\_issued()

print('-------------------------------------------------')

elif choice == 'R':

return\_book()

print('-------------------------------------------------')

elif choice == 'B':

book\_issued()

print('-------------------------------------------------')

elif choice == 'D':

print(structure)

print('-------------------------------------------------')

elif choice == 'X':

delete\_book()

print('-------------------------------------------------')

elif choice == 'V':

available()

print('-------------------------------------------------')

elif choice == 'S':

structure.to\_csv('Book1.csv', index=False)

count = count + 1

elif choice == 'Q':

if count == 0:

print("You have not saved your work. Do you want to exit anyways?:")

decision = input(">")

if decision == 'Yes':

break

else:

continue

else:

break

else:

print('-------------------------------------------------')

print('Invalid choice, Choose again')

print('-------------------------------------------------')

print("Good Bye")