Library Management System Assignment Tables Information:-

1. **Users Table (to maintain the registered user authentication details)**

UserId	int (Primary Key)
UserName	varchar(50)
Password	varchar(50)

2. **Books Table (to maintain the book details in the library)**

Book_Id	int (Primary Key)
Book_title	varchar(120)
Author_id	int (Uses the author_id field of Author table)
Category_id	int (Uses the category_id field of Category table)
Book_checkedout	Bit (False when book available in library i.e not checkedout)

3. **Category Table (to maintain the category details)**

Category_id	int (Primary Key)
Book_category	varchar(80) (name of the category)

4. **Author Table (to maintain the author details)**

author_id	int (Primary Key)
Book_author	varchar(80) (name of the author)

5. **UpdateTransactions Table(to maintain the logs of borrow and return of books)**

User_id	int (Foreign Key from Users table)
Book_id	int (Foreign Key from Books Table)
Transaction_type	varchar(25) (checkin/checkedout)
Transaction_date	date

6. **BorrowTransactions Table (to maintain the records of borrowed books)**

User_id	int (Foreign Key from Users table)
Book_id	int (Foreign Key from Books Table)
Transaction_date	date
Due_date	date

7. **EditTransactions Table (to maintain the logs of edits done to the category, author, book details)**

Book_id	int (Null allowed when edit not done with respect to one book)
User_id	int
Modified_field	varchar(25)
Old_value	varchar(80)
New_value	varchar(80)
Edit_date	date

Library Management System Assignment Functionalities and the required tables it is using:-

Sr. No.	Request	PathInfo	Tables Used
1	AllAuthors(Get)	/authors	**Used:- <mark>Authors table</mark>**
			Authors (To feth name of authors along with their id)
2	Addauthor(Post)	/authors	**Used:- <mark>Authors</mark>**
			Authors (To check whether the author already exists)
			Authors (To add name of the new author along with their id)
3	EditAuthor(Put)	/authors	**Used:- <mark>Authors, EditTransactions table</mark>**
			Authors (To check whether the author id is valid or not)
			Authors (To store the present value of book_author before the edit, to update in EditTransactionstable)
			Authors (To check whether the proposed edit is same as the current author name)
			EditTransactions (to insert the record of edit**)**
4	AllBooks (Get)	/book	**Used:- <mark>Books, Category, Author</mark>**
			Books (To fetch all details of book)
			Author (To fetch author name from the author_id in books table)
			Category (To fetch the category name from the category_id from books table)
5	AddBooks(Post)	/book	**Used:- <mark>Books, Category, Author</mark>**
			Books (To check if entered author_id and category_id is valid)
			Books (To check if entered book title already exist)
			Author (To fetch author name from the author_id in books table)
			Category (To fetch the category name from the category_id from books table)
6	SearchBooksAccordingToFilter(Get)	/bookfilter	**Used:- <mark>Books</mark>**
			Books (To Get details of books in JSON list according to given filter field in string format)

7	Login (Post)	/login	**Used:-** <mark>**Users**</mark>
			Users (To check whether the user exist in database)
8	BorrowBook(Post)	/borrow	**Used:-** <mark>**BorrowTransactions, Books, UpdateTransactions**</mark>
			BorrowTransactions (To check whether the user has borrowed two books (user cannot borrow more than 2 books without returning previously borrowed books))
			Books (To check whether the requested book exist in the library and if it is available (not checked out) in the library)
			UpdateTransactions(To add record of borrow)
			Books (To change the checked_out status of book to True)
9	ReturnBook(Post)	/return	**Used:-** <mark>**BorrowTransactions, UpdateTransactions**</mark>
			BorrowTransactions (To check whether the book is borrowed, to be returned (does'nt need to be the same user who borrowed the particular book anyone can return))
			UpdateTransactions(To add record of return)
			Books (To change the checked_out status of book to False)
10	GetCategory(Get)	/category	**Used:-** <mark>**Category**</mark>
			Category(To fetch all Category details)
11	AddCategory (Post)	/category	**Used:-** <mark>**Category**</mark>
			Category (To check whether category already exist)
			Category (To add name of the new category along with their id)
12	EditCategory(Post)	/category	**Used:-** <mark>**Category, EditTransactions table**</mark>
			Category (To check whether the category_id is valid or not)
			Category (To store the present value of book_category before the edit, to update in EditTransactionstable)
			Category (To check whether the proposed edit is same as the current category name)
			EditTransactions (to insert the record of edit**)**
13	GetTransactionHistory(Get)	/transaction history	**Used:-** <mark>**UpdateTransactions**</mark>
			UpdateTransactions(To fetch all Transaction details of the logged in user)
14	GetEditTransactions (Get)	/patients	**Used:-** <mark>**EditTransactions**</mark>
			EditTransactions(To fetch Edit Transaction details)