

Library Management System

**Dept. Of Computer Science
Illinois Institute of Technology
Database Organization
CS-425 (Group-25)
Instructor: Yousef Elmehdwi**

**By
Anusha SP (A20518694)
Sneha Rajshekhar Hiremath (A20514161)
Swetha Sighakolli (A20516677)**

DOCUMENT HISTORY:

Version	Date	Author	Description of change
1	30 Sep 2022	Group-25	ER diagram added
2	05 Nov 2022	Group-25	Relational schema and Create script added
3	9 Dec 2022	Group 25	Insert script added
4	10 Dec 2022	Group-25	Implementation details and UI screenshots added.

ABSTRACT

This project revolves around a library management system. Two types of users are considered: Members and Librarians. Members can search for documents, borrow documents, and return documents. Documents can be borrowed for certain time periods. If a member does not return a document by the due date, he/she will be notified. Librarians can check overdue documents, manage users, and add/delete/modify documents. The main asset of the library are documents. A document can be a book/journal/magazine. There will be a publisher for each document type. We are logging all the activities performed by members and librarians in an activity log table so that it would be easy to debug in case of any issues.

The application supports below main functionalities:

- Members and Librarians can search for documents.
- Members can return/borrow the documents.
- Librarians can add/delete/modify the document.
- Members will be notified of the document due dates.

In addition to the above main functionalities, few other functionalities we added are as below:

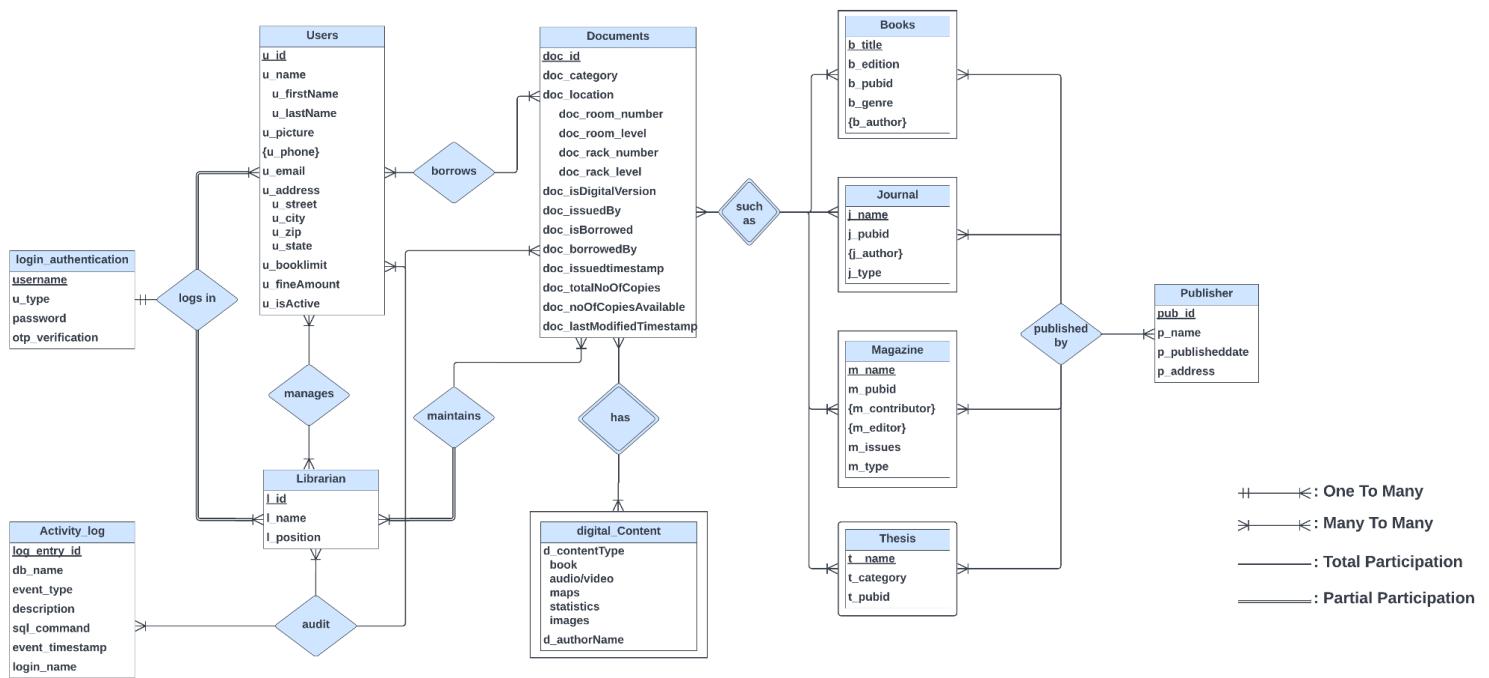
- Thesis and Digital content are the additional document types added.
- Ranking search results according to the match criteria.
- Showing whether digital content is available for each document/not.

Table of Contents

1. Entity Relationship Diagram	5
2. Create script	6
3. Relational schema	7
3.1 Strong entities and Composite attributes	7
3.2 Weak entities	8
3.3 Multi-valued attributes	9
3.4 Relationships	10
4. Insert script	13
5. Implementation details	14
6. User interface screens.....	15
7. Acknowledgement	24

1. ENTITY RELATIONSHIP DIAGRAM

LIBRARY MANAGEMENT SYSTEM ER DIAGRAM



2.CREATE SCRIPT

We have created the entities below.

- Login_auth
- Documents
- Activity_log
- Librarian
- Users
- Users_phone
- Publisher
- Digital content
- Books
- Book_authors
- Journal
- Journal_authors
- Magazine
- Magazine_editors
- Thesis
- Logs_in
- Manages
- Maintains
- Borrows
- Published_by

The create script for them is attached below:

[Phase2_CreateScript.txt](#)

3. RELATIONAL SCHEMA

3.1. STRONG ENTITIES AND COMPOSITE ATTRIBUTES:

Login_authentication(username, u_type, password_otp_verification)

Activity_log(log_entry_id, db_name, event_type, description, sql_command, event_timestamp, login_name)

Librarian(l_id, l_name, l_position)

Users(u_id, u(firstName, u.lastName, u.picture, u_email, u_street, u_city, u_zip, u_state, u_booklimit, u_fineAmount, u_isActive)

Documents(doc_id, doc_category, doc_room_number, doc_room_level, doc_rack_number, doc_rack_level, doc_isDigitalVersion, doc_issuedBy, doc_borrowedBy, doc_issuedTimestamp, doc_totalNoOfCopies, doc_noOfCopiesAvailable, doc_lastModifiedTimestamp)

Publisher(pub_id, p_name, p_publisheddate, p_address)

3.2 WEAK ENTITIES:

Login_authentication(username, utype, password_otp_verification)

Activity_log(log_entry_id, db_name, event_type, description, sql_command, event_timestamp, login_name)

Librarian(l_id, l_name, l_position)

Users(u_id, u(firstName, u.lastName, u.picture, u_email, u_street, u_city, u_zip, u_state, u_booklimit, u_fineAmount, u_isActive)

Documents(doc_id, doc_category, doc_room_number, doc_room_level, doc_rack_number, doc_rack_level, doc_isDigitalVersion, doc_issuedBy, doc_borrowedBy, doc_issuedTimestamp, doc_totalNoOfCopies, doc_noOfCopiesAvailable, doc_lastModifiedTimestamp)

Publisher(pub_id, p_name, p_publisheddate, p_address)

DigitalContent(doc_id, book, audioVideo, maps, statistics, images, d_authortname)

Books(doc_id, b_title, b_edition, b_pubid, b_genre)

Journal(doc_id, j_name, j_pubid, j_type)

Magazine(doc_id, m_name, m_pubid, m_issues, m_type)

Thesis(doc_id, t_name, t_category, t_pubid)

3.3 MULTI VALUED ATTRIBUTES:

Login_authentication(username, utype, password_otp_verification)

Activity_log(log_entry_id, db_name, event_type, description, sql_command, event_timestamp, login_name)

Librarian(l_id, l_name, l_position)

Users(u_id, u(firstName, u.lastName, u.picture, u_email, u_street, u_city, u_zip, u_state, u_booklimit, u_fineAmount, u_isActive)

Users_phone(u_id,u_phone)

Documents(doc_id, doc_category, doc_room_number, doc_room_level, doc_rack_number, doc_rack_level, doc_isDigitalVersion, doc_issuedBy, doc_borrowedBy, doc_issuedTimestamp, doc_totalNoOfCopies, doc_noOfCopiesAvailable, doc_lastModifiedTimestamp)

Publisher(pub_id, p_name, p_publisheddate, p_address)

DigitalContent(doc_id, book, audioVideo, maps, statistics, images, d_authortname)

Books(doc_id, b_titile, b_edition, b_pubid, b_genre)

Book_authors(doc_id, b_titile,b_author)

Journal(doc_id, j_name, j_pubid, j_type)

Journal_authors(doc_id, j_name,j_author)

Magazine(doc_id, m_name, m_pubid, m_issues, m_type)

Magazine_Contributors(doc_id, m_name,m_contributor)

Magazine_Editors(doc_id, m_name,m_editor)

Thesis(doc_id, t_name, t_category, t_pubid)

3.4 RELATIONSHIPS:

Login_authentication(username, u_type, password_otp_verification)

Activity_log(log_entry_id, db_name, event_type, description, sql_command, event_timestamp, login_name, doc_id)

Librarian(l_id,l_name, l_position)

Users(u_id, u(firstName, u.lastName, u.picture, u.email, u.street, u.city, u.zip, u.state, u.booklimit, u.fineAmount, u.isActive)

Users_phone(u_id,u_phone)

Documents(doc_id, doc_category, doc_room_number, doc_room_level, doc_rack_number, doc_rack_level, doc_isDigitalVersion, doc_issuedBy, doc_borrowedBy, doc_issuedTimestamp, doc_totalNoOfCopies, doc_noOfCopiesAvailable, doc_lastModifiedTimestamp)

Publisher(pub_id, p_name, p_publisheddate, p_address)

DigitalContent(doc_id, book, audioVideo, maps, statistics, images, d_authorname)

Books(doc_id, b_titile, b_edition, b_pubid, b_genre)

Book_authors(doc_id, b_titile, b_author)

Journal(doc_id, j_name, j_pubid, j_type)

Journal_authors(doc_id, j_name, j_author)

Magazine(doc_id, m_name, m_pubid, m_issues, m_type)

Magazine_Contributors(doc_id, m_name, m_contributor)

Magazine_Editors(doc_id, m_name, m_editor)

Thesis(doc_id, t_name, t_category, t_pubid)

Logs_in(u_id, l_id) //Many to Many relationship

Manages(u_id, l_id) //Many to Many relationship

Published_By(doc_id,b_title,j_name,m_name,t_name, pub_id) //Many to Many relationship

Maintains(l_id, doc_id) //Many to Many relationship

Borrows(u_id, doc_id) //Many to Many relationship

4. INSERT SCRIPT

The insert script for inserting sample data into the database for testing purposes is attached below.

It records 5 sample records into each table.

[InsertData_Script.txt](#)

5. IMPLEMENTATION DETAILS

The implementation details are as below:

- APIs are developed in spring boot using the JPA repository.
- The above developed APIs are integrated with ReactJS library based UI.
- Three different UI components have been developed for different dashboard views to different members.
 1. Login page view
 2. Librarian dashboard
 3. Member dashboard
- The authentication used is Spring security bearer token generation. This token is generated by the server in response to the login request. For subsequent requests, the client must send this token in the authorization header. This is for secure transmission of information.

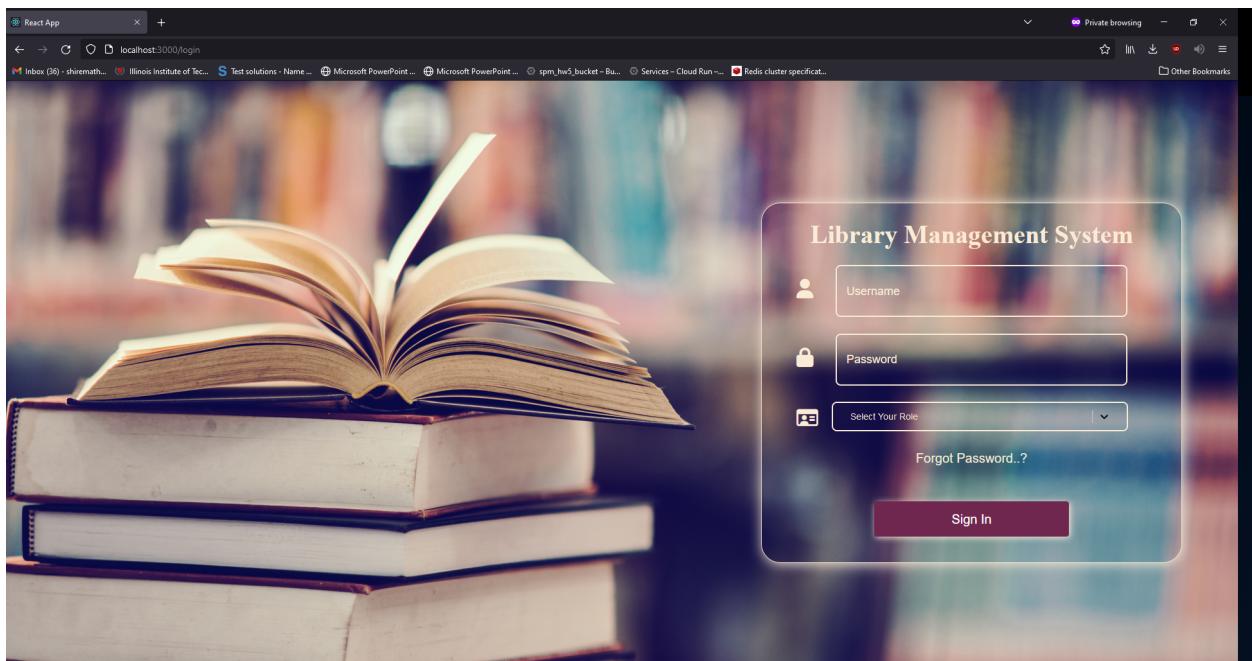
GIT URL OF THE SOURCE CODE:

<https://github.com/Snehahiremath24/library-management-system>

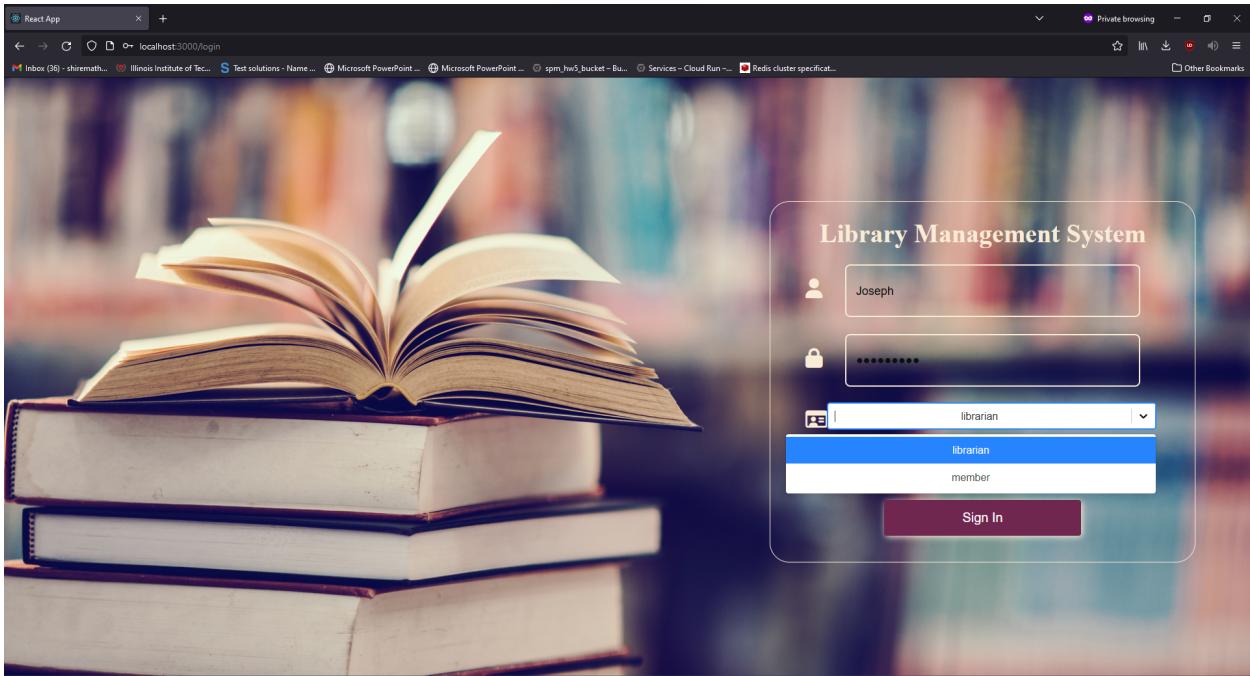
6. USER INTERFACE SCREENS

SCREEN 1: Login Screen

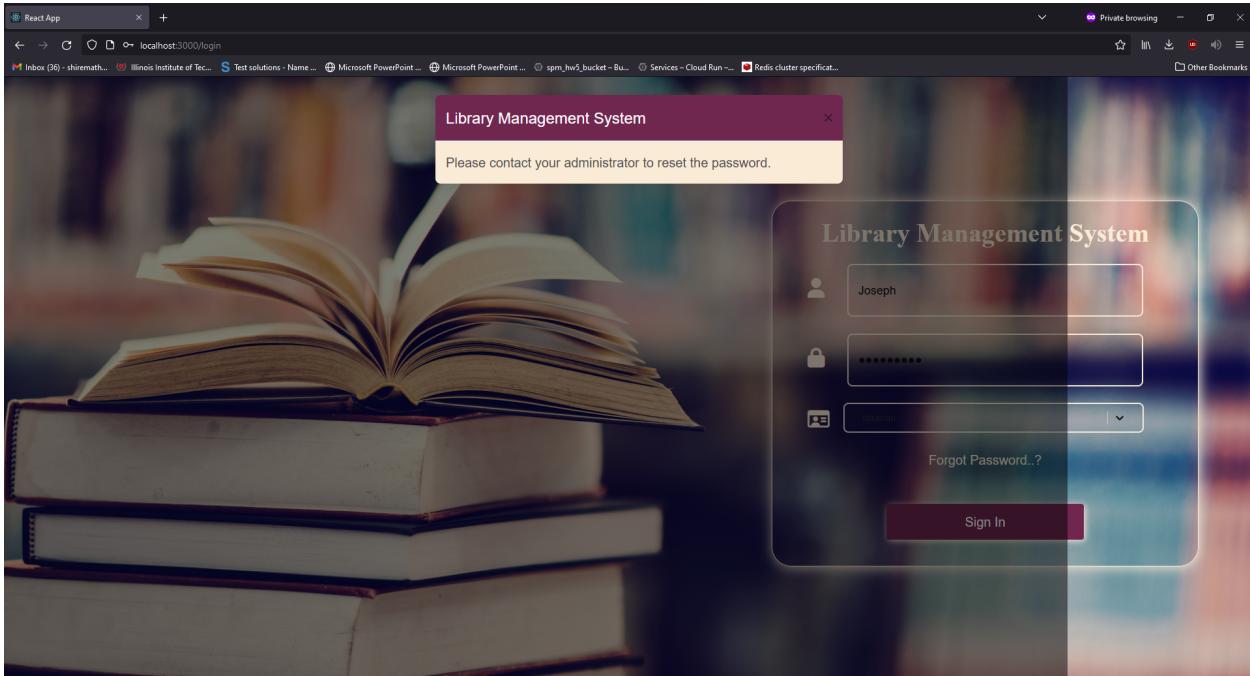
- This screen is common for both member login and librarian login.
- It takes three input parameters: username, password and user type: member/librarian.
- When a user clicks on submit, the password is encrypted and validated against the password in the database using Bcrypt encoding of java and a token is generated.
- This generated token will be passed in the subsequent requests making the API calls secure.



The dropdown for selecting the usertype is as below:

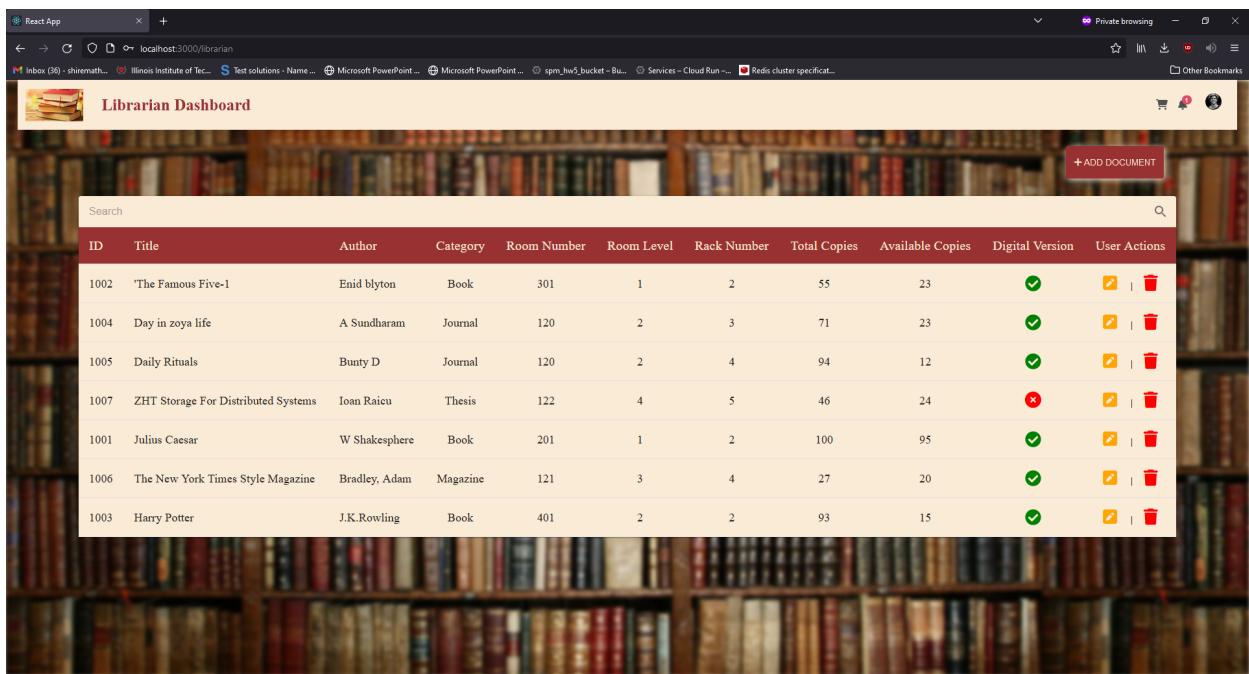


SCREEN 2: If username clicks on 'Forgot Password':



SCREEN 3: Librarian Dashboard

- This is a librarian homescreen.
- A librarian can add/delete/edit documents and search for documents.
- The home screen shows several fields like ID, Title, Author, Category, Room number, Room level, Rack number, Total copies, Available copies, digital content and the necessary actions.

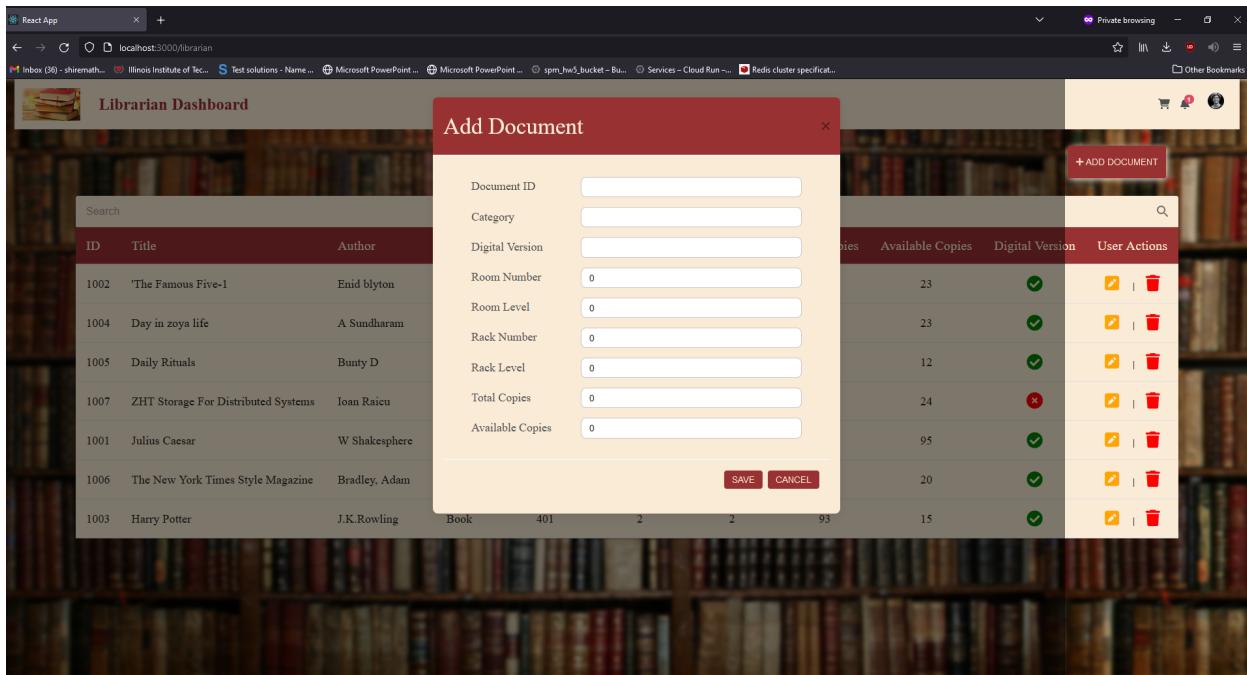


The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/librarian". The page has a header with a book icon and the text "Librarian Dashboard". Below the header is a search bar and a table with the following data:

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	User Actions
1002	The Famous Five-1	Enid blyton	Book	301	1	2	55	23		
1004	Day in zoya life	A Sundharam	Journal	120	2	3	71	23		
1005	Daily Rituals	Bunty D	Journal	120	2	4	94	12		
1007	ZHT Storage For Distributed Systems	Ioan Raiucu	Thesis	122	4	5	46	24		
1001	Julius Caesar	W Shakespeare	Book	201	1	2	100	95		
1006	The New York Times Style Magazine	Bradley, Adam	Magazine	121	3	4	27	20		
1003	Harry Potter	J.K.Rowling	Book	401	2	2	93	15		

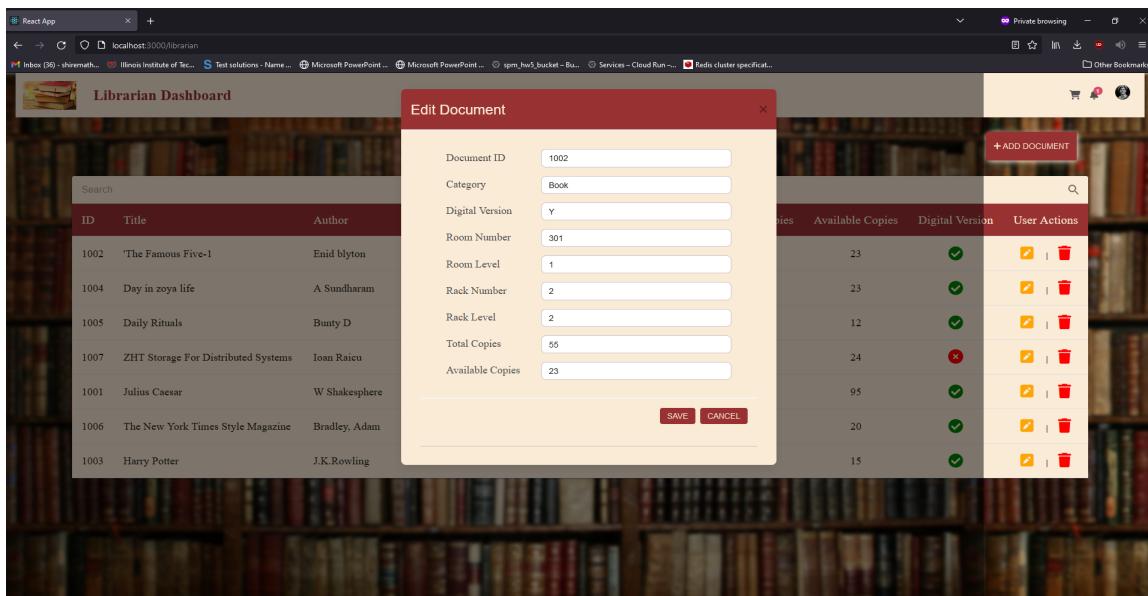
SCREEN 4: Add document screen

- A librarian can document data into the database.
- When he inputs required fields and clicks on Submit, the data will be inserted and will be displayed on the dashboard home screen immediately within a span of 1 second.



SCREEN 5: Edit document screen

- A librarian can edit existing documents in the database.
- When he inputs required fields and clicks on Submit, the data will be updated and will be displayed on the dashboard home screen immediately within a span of 1 second.



SCREEN 6: Delete document screen

- A librarian can delete existing documents in the database.
- When the librarian clicks on delete, the record will be deleted from the dashboard and will no longer be displayed.

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	User Actions
1002	'The Famous Five-1	Enid blyton	Book	301	1	2	55	23		
1004	Day in zoya life	A Sundharam	Journal	120	2	3	71	23		
1005	Daily Rituals	Bunty D	Journal	120	2	4	94	12		
1007	ZHT Storage For Distributed Systems	Ioan Raicu	Thesis	122	4	5	46	24		
1001	Julius Caesar	W Shakesphere	Book	201	1	2	100	95		
1006	The New York Times Style Magazine	Bradley, Adam	Magazine	121	3	4	27	20		
1003	Harry Potter	J.K.Rowling	Book	401	2	2	93	15		

SCREEN 7: Search by title

- The search field supports search by ID, title, author and category.
- Partial search is also supported.
- Results are ranked and listed as per their match criteria.

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	User Actions
1003	Harry Potter	J.K.Rowling	Book	401	2	2	93	15		

SCREEN 8: Partial search

A screenshot of a web browser window titled "React App" showing the "Librarian Dashboard". The dashboard features a background image of bookshelves. A modal window is open, displaying a table of library documents. The table has columns: ID, Title, Author, Category, Room Number, Room Level, Rack Number, Total Copies, Available Copies, Digital Version, and User Actions. The search term "d" is visible in the modal's header. The table contains the following data:

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	User Actions
1002	The Famous Five-1	Enid blyton	Book	301	1	2	55	23	✓	✖ 🗑
1004	Day in zoya life	A Sundharam	Journal	120	2	3	71	23	✓	✖ 🗑
1005	Daily Rituals	Bunty D	Journal	120	2	4	94	12	✓	✖ 🗑
1007	ZHT Storage For Distributed Systems	Ioan Raicu	Thesis	122	4	5	46	24	✗	✖ 🗑
1006	The New York Times Style Magazine	Bradley, Adam	Magazine	121	3	4	27	20	✓	✖ 🗑

SCREEN 9: Full search

A screenshot of a web browser window titled "React App" showing the "Librarian Dashboard". The dashboard features a background image of bookshelves. A modal window is open, displaying a table of library documents. The table has columns: ID, Title, Author, Category, Room Number, Room Level, Rack Number, Total Copies, Available Copies, Digital Version, and User Actions. The search term "daily" is visible in the modal's header. The table contains the following data:

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	User Actions
1005	Daily Rituals	Bunty D	Journal	120	2	4	94	12	✓	✖ 🗑

SCREEN 10: Member Dashboard

- This is a member homescreen.
- A member can return/borrow documents and search for documents.
- The home screen shows several fields like ID, Title, Author, Category, Room number, Room level, Rack number, Total copies, Available copies, digital content, issue date, due date and the necessary actions.
- It also shows a notification icon where it displays if a book is nearing due or is past due date. (7 days is considered as the deadline for return)

The screenshot shows a web browser window titled "React App" displaying the "Member Dashboard". The URL is "localhost:3000/members". The page features a header with a logo and navigation links. Below the header is a search bar labeled "Search for documents and more". The main content is a table with the following columns: ID, Title, Author, Category, Room Number, Room Level, Rack Number, Total Copies, Available Copies, Digital Version, Borrow | Return, Issue Date, Due Date, and a bell icon for notifications. The table contains the following data:

ID	Title	Author	Category	Room Number	Room Level	Rack Number	Total Copies	Available Copies	Digital Version	Borrow Return	Issue Date	Due Date	Notification
1002	'The Famous Five-1	Enid blyton	Book	301	1	2	55	23	✓	↕	-	-	-
1004	Day in zoya life	A Sundharam	Journal	120	2	3	71	23	✓	↕	-	-	-
1005	Daily Rituals	Bunty D	Journal	120	2	4	94	12	✓	↕	-	-	-
1007	ZHT Storage For Distributed Systems	Ioan Raicu	Thesis	122	4	5	46	24	✗	↕	-	-	-
1001	Julius Caesar	W Shakespeare	Book	201	1	2	100	95	✓	↕	Wed Nov 09 2022	Wed Nov 16 2022	Past Due Date
1006	The New York Times Style Magazine	Bradley, Adam	Magazine	121	3	4	27	20	✓	↕	Fri Dec 09 2022	Fri Dec 16 2022	Due Date Is Coming
1003	Harry Potter	J.K.Rowling	Book	401	2	2	93	15	✓	↕	Fri Dec 09 2022	Fri Dec 16 2022	Due Date Is Coming

SCREEN 11: Return document Success

- If the member clicks on Return on any document, the available count of that document is incremented.

SCREEN 12: Return document Failure

- If the member clicks on Return on the document he/she didn't borrow, below error message would be displayed.

SCREEN 13: Borrow document

- If the member clicks on Borrow on the document, the available count will be decremented and issue date, due date and notification of due will be displayed.

*The alert messages for above three functionalities were displayed only for a second on screen. Was difficult to capture. Will be shown as part of the demo.

7. ACKNOWLEDGEMENT

We would like to give a special “Thank You” to our professor Yousef Elmehdwi and the teaching assistants Mohammadreza Sediqin and Shraddhaben Patel who provided incredibly valuable guidance throughout this project.

****DRIVE LINK OF ALL THREE PHASES OF SUBMISSION AND DEMO:**

The below drive link contains:

- Phase 1: ER diagram
- Phase 2: Create script
- Phase 3: Source code zip + Demo of the working application
- Insert script for testing the application
- Project report

https://drive.google.com/drive/folders/1XASKgW6-OIAb5VpAdTIX2qrFh4Md_pt