

CS 432: DATABASES

LIBRARY MANAGEMENT SYSTEM



ASSIGNMENT 4

(Datasaurus)

GROUP MEMBERS

Aashmun Gupta	22110005
Anmol Kumar	22110028
Anushri Sanodia	22110030
Deepanjali Kumari	22110069
Dhruv Sharma	22110074
Pavani Khale	22110191
Yash Patel	21110243

Under the Guidance of
Prof. Mayank Singh

Responsibility of G1:

1. Changes suggested by the stakeholders :

- To show the availability of the books in the online catalogue model.
- Faculty recommendation - The catalogue recommended by the faculty should be added by the admin. The faculty can only suggest the catalogue thus needing a suggestion table.
- Details of the book - While cataloguing, they suggested we add more information about the book.
- While searching for a particular catalogue, the drop down should have various options like search by name, author, publisher, genre, duration and many more.
- Generating the return date of a catalogue after issuing and then connecting the mail and calendar together to send the reminder mail.
- Penalty record based on the number of days passed and the total penalty the user must pay.
- Generation of reports - The library generates reports on a daily basis depending upon the requirements, which include - the number of books issued, total users enrolled, collection of a particular day based on the books not returned and many more. To generate these in csv format for easy viewing and editing.
- To import the bulk data from excel in case of catalogue and users.

BEFORE:

ADD CATALOGUE

ADD MEMBER

ADD CATEGORY

Catalogue List

Title	Catalogue ID	Category No	Type	Author	Publisher	Count	Actions
Koi Deewana Kehta Hai	6	1	Book	Kumar Vishwas	Raj Kamal	100	<div>UPDATEDELETE</div>
Musafir Hun	5	1	Book	Rahat Indori	Raj Ramal	30	<div>UPDATEDELETE</div>

Admin Page

Catalogue Search

Search Results for "musa"

Musafir Hun

Author: Rahat Indori
Publisher: Raj Ramal
Material Type: Book

Catalogue Search Available

AFTER:



Search
musafir

All

All

Books

Journals

Magazines

Your Summary

CHECKED OUT

YOUR REQUESTS

Catalogue Search By Catagories

Faculty Page

ADD RECOMMENDATION

Your Recommend List

User ID	Catalogue ID	Course ID	Actions
113	6	ES 221	<div>UPDATEDELETE</div>

Added Faculty Recommendation System

Report Generation

GENERATE CATALOGUE REPORT

GENERATE BOOKS ISSUED REPORT

Report Generation UI

	A	B	C	D	E	F	G	H	I	J
1	Catalogue ID	Catalogue Title	Issued By	Issue Date						
2		5 Musafir Hun	fsweett3	31-03-2024						
3		5 Musafir Hun	vjeger2	14-02-2024						
4		6 Koi Deewana Kehta Hai	fsweett3	09-04-2024						
5										
6										
7										
8										
9										
10										

Excel Reports of Library are available

Admin View:

Administrators have complete control over the library management system. They can perform the following operations:

- **Catalog Management:** Admins can add, edit, and delete books from the library catalog.
- **User Management:** They have the authority to add, remove, and modify user accounts, including students, faculty, and other staff.
- **Transaction Management:** Admins can oversee all library transactions, including issuing, returning, and renewing books.
- **Reporting:** They can generate various reports such as transaction history, inventory status, and user activity.

HomeAboutLMS ProjectManage External LibraryCourse Recommendation

ADD CATALOGUE

ADD MEMBER

ADD CATEGORY

Catalogue List

Title	Catalogue ID	Category No	Type	Author	Publisher	Count	Actions
<h2>Fluids</h2>	10	2	Book	MC.Hill	qwerty	500	<div>UPDATEDELETE</div>
Gunaho ka Dewta	9	2	Book	Mohan Rakesh	Raj Kamal	50	<div>UPDATEDELETE</div>
Koi Deewana Kehta Hai	6	1	Book	Kumar Vishwas	Raj Kamal	100	<div>UPDATEDELETE</div>
Musafir Hun	5	1	Book	Rahat Indori	Raj Ramal	30	<div>UPDATEDELETE</div>

Faculty View:

Faculty members can recommend books to be added to the library collection, especially those relevant for course readings.

HomeAboutLMS Project

ADD RECOMMENDATION

Your Recommend List

User ID	Catalogue ID	Course ID	Actions
113	6	ES 221	<div>UPDATEDELETE</div>

Student View:

Students can search for books in the library catalogue based on title, author, subject, or keyword. They can also perform the following:

- Issue: They can borrow books from the library by checking them out using their student ID.
- Return: Students can return borrowed books to the library after they have finished using them.
- Renewal: Depending on the library policies, students may be allowed to renew their borrowed books for an extended period.

Responsibility of G2:

1. Concurrent multi-user access:

To implement this, we have done role-based preventative measures, which only allows specific tasks to be performed by specific users. Additionally, We have implemented SQL Table Locks to allow only a single transaction at a time. We have taken care of this while working on UPDATE and INSERT Commands.

We used the following code to lock the tables. For example,

```
# Update catalogue
cur.execute("LOCK TABLES catalogue WRITE")
```

After the catalogue is updated, we again unlock the table

```
cur.execute("UNLOCK TABLES")
```

With this code we tried to update the catalogue,

Update Catalogue

Catalogue ID

12

Title

India The Great

Author Name

Jaishankar

Update Catalogue

Catalogue ID

12

Title

India The Great Country

Author Name

Jaishankar

Publisher Name

Title	Catalogue ID	Category No	Type	Author	Publisher
India The Great Country	12	5	Journal	Jaishankar	Penguin

The Data is updated successfully with the LOCK mechanism

2.

The changes that have been made to the database are as follows:

a) Making catalogue_id AUTO_INCREMENT

We were asked to reduce the manual entering of catalogue ID every time a new catalogue is added. This change simplifies the process of inserting new records into the table, as the database system handles the assignment of primary keys without requiring manual intervention.

b) Adding category_name to the shelf table:

The stakeholders asked us to assist them in choosing the category number to which the catalogue belongs. Users will only have to choose the category name from any dropdown to reduce manual entering of category number

c) Changing user_ID to int auto_increment

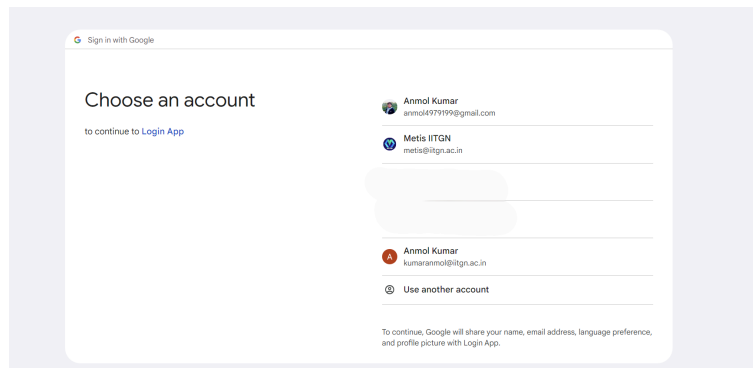
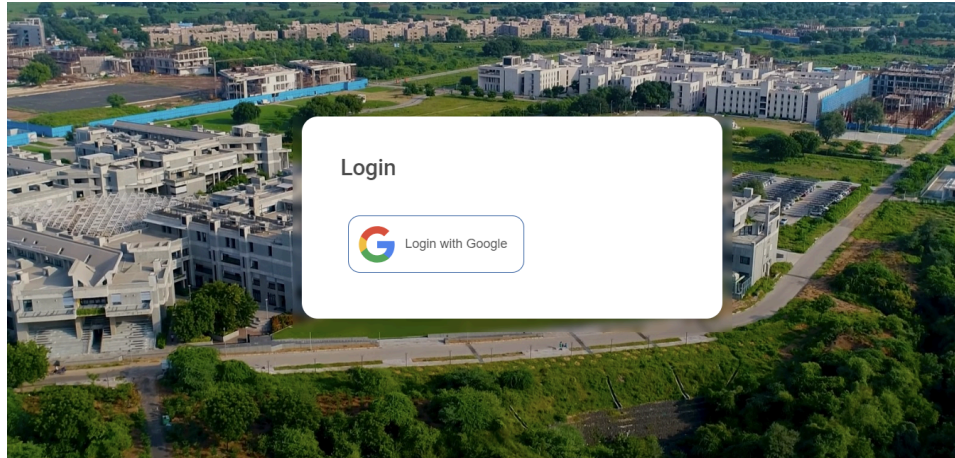
Changing user_ID to an auto-incrementing integer primary key simplifies user management and ensures each user has a unique identifier automatically assigned by the database system.

3.

We have implemented a Google login system to secure access to the library resources. The login process uses the OAuth 2.0 protocol to authenticate users. By using the Google login, we can detect the type of user accessing the system.

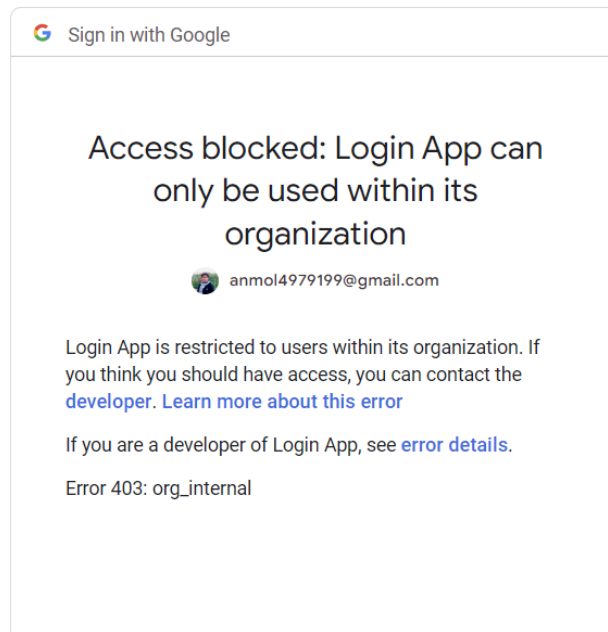
The **user** table in our database has a **member_type** column that stores the user's role or access level. After a user successfully logs in with their Google account, we check if the domain of their email address is "iitgn.ac.in". If it is, we retrieve the email address from the user_mail table and compare it to the authenticated email address.

If the user is found in the system, they are then redirected to their specific views based on their assigned member type. This ensures that only authorized users from the IITGN domain can access the library resources, and their access is tailored to their roles and permissions.



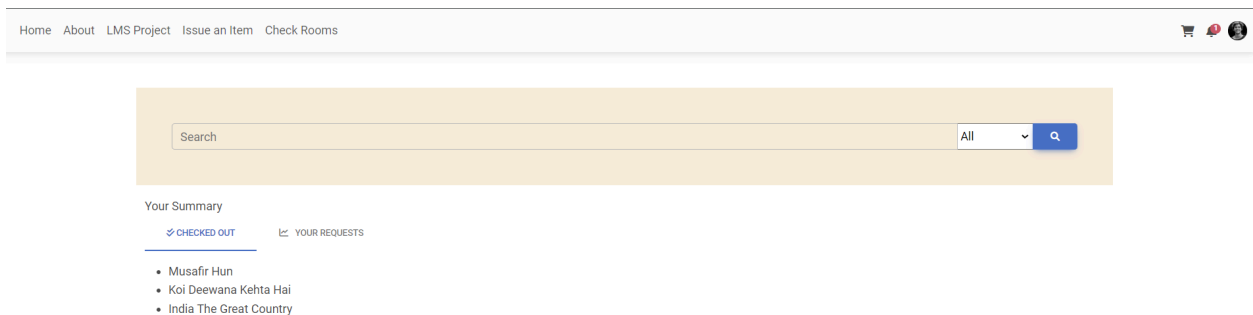
Login Screen of Google

After we select the non-IITGN user, it gives an error



Error for anmol4979199@gmail.com

For all IITGN users, the specific views for them will be opened.

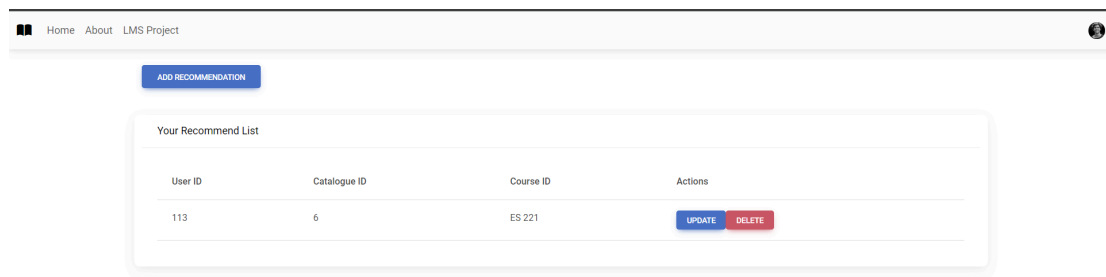
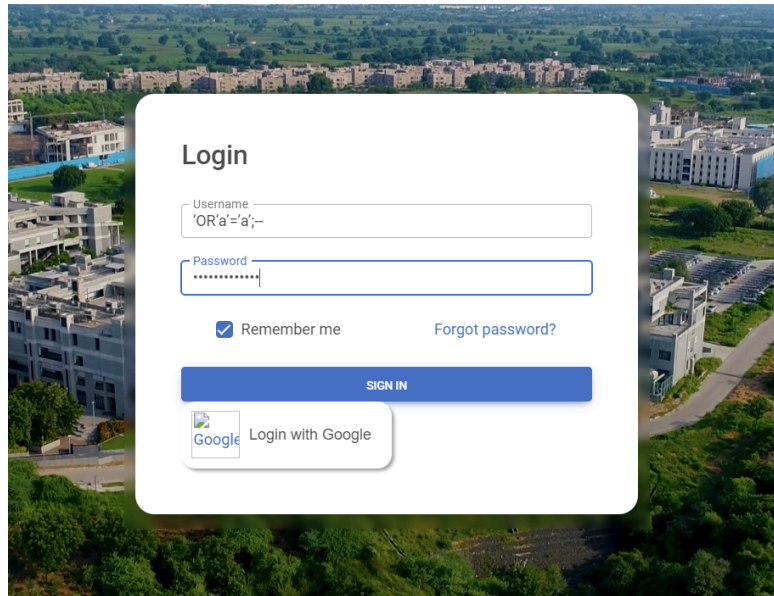


Responsibility of G1 and G2:

1.

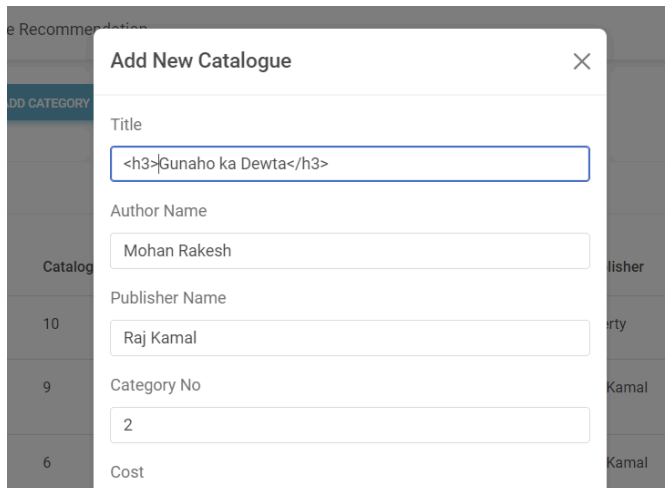
a) SQL Injection Attack

We performed SQL Injection in the Login Page where we entered 'OR'a='a';-- to check if the SQL attack worked. With this code we got logged into the admin view. We fixed this issue by utilizing the inbuilt parameterised input of the Flask feature. Now, all the inputs will be treated as strings and they will not execute in SQL queries.



b) XSS Attack -

We attempted an XSS attack by injecting HTML code into the catalogue table and found that the characters were not being escaped. Therefore, we implemented the Bleach library to clean and sanitize the data before adding it to the database.



The screenshot shows a modal window titled "Add New Catalogue" with a close button (X). The form contains the following fields:

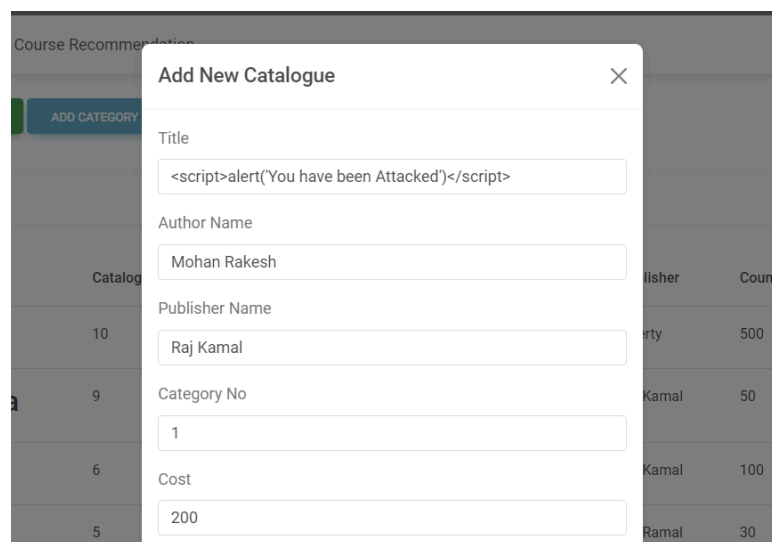
- Title: `<h3>Gunaho ka Dewta</h3>`
- Author Name: Mohan Rakesh
- Publisher Name: Raj Kamal
- Category No: 2
- Cost: (empty)

The Result:

Gunaho ka Dewta	9	2	Book	Mohan Rakesh	Raj Kamal	50	<button>UPDATE</button> <button>DELETE</button>
Koi Deewana Kehta Hai	6	1	Book	Kumar Vishwas	Raj Kamal	100	<button>UPDATE</button> <button>DELETE</button>
Musafir Hun	5	1	Book	Rahat Indori	Raj Ramal	30	<button>UPDATE</button> <button>DELETE</button>

This means that the code in HTML was getting executed.

We tried with other tests like adding a javascript code in the form. We came to know that the javascript code was also getting executed.



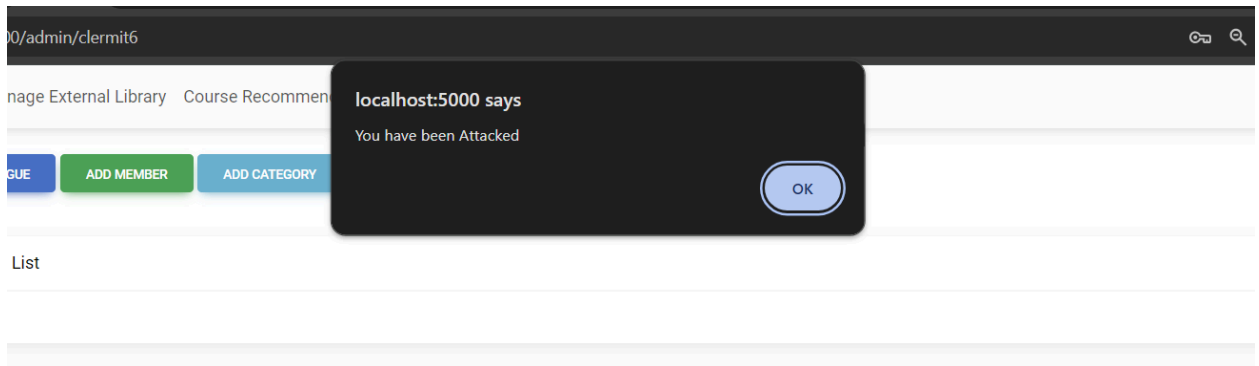
The screenshot shows the same "Add New Catalogue" modal window. The title field now contains the following JavaScript code:

`<script>alert('You have been Attacked')</script>`

The other fields remain the same:

- Author Name: Mohan Rakesh
- Publisher Name: Raj Kamal
- Category No: 1
- Cost: 200

(Malicious Code)



The code gets executed.

After We sanitized the data and HTML encoded it, no special character could be executed on our system. **Here is the screenshot after the change**

Catalogue List				
Title	Catalogue ID	Category No	Type	Author
<script>alert("You have been Attacked")</script>	11	1	Book	Mohan
<h2>Fluids</h2>	10	2	Book	MC.Hill

c) Brute Force Attack

In our web app, we have allowed users to log in using their username and password. This can raise the possibility of brute-force attacks To tackle with this issue, we implemented a timer and IP address-based check for users. If a user logs in incorrectly for five consecutive times within 30 seconds, the system will assume that a brute force attack is being tried on it. It will restrict access to login routes.



Too many login attempts. Please try again later.

2. Show that all the relations and their constraints, finalized after the second feedback, are present and valid as per the ER diagram constructed in Assignment 1.

The ER Diagram is attached here for your reference: [file](#)

The ER diagram represents all the tables we made but during the second assignments, we merged some of them together into tables. The information has been attached while uploading the second assignment file.

So our final tables are:

1. User:

This table stores information about all types of members registered within the IITGN library community. The website accesses the information from here and gives them view access to the pages depending upon the member type. Hence, the user table has one of the strongest utilities as needed for the functioning of the website.

2. User_mail:

This table is strongly created to store email addresses as this has specific uses but connected through a foreign key in the user table. This allows us to put more than one mail address associated with a particular user. We can add user mail for those users only who have their details in the user table.

3. Catalogue:

This table is merged with all the tables like books, DVDs, magazines and newspapers. They are in disjoint sets and hence we combined all of them as shown in ER. We used this to search any type of category in the catalog tables and display results henceforth.

4. External_library:

It stores data for other libraries that play a part in the exchange of resources. We can keep track of its data.

5. Author:

This table is created to keep separate sections of authors so that we don't create a different section while creating a separate room for books of authors again. Such that if it's already present we can directly modify any changes and search for it. The author id from here is used in the catalogue table as a foreign key. Thus while cataloging if the author name already exists it directly takes its id from the authors table.

6. Publisher:

This table is created to keep separate sections of publishers such that we keep a different section while creating a separate list for publishers we purchase from. Such that if it's already present we can directly modify any changes and search for it. The publisher id from here is used in the catalog table as a foreign key. Thus while cataloging if the publisher name already exists it directly takes its id from the publishers table.

7. Shelf:

This table shows the shelf location of catalogues, We added category_name as a new column as per the received feedback. It stores the subject of the catalogue.

8. Issue:

The Issue table stores the issue_id and Issue_date of the transaction. The issue_id is essential because it is used in different tables like issuing, etc. The issue id is the unique id.

9. Event:

It gives access only to admin so that they can make changes in all the basic information like location, availability or other information that others can only view.

10. InterLibrary:

This table contains information for people other than IITGN college accessing library services.

11. Issuing:

Gives information about which catalogue is issued with its issue id and id associated with the user.

12. Lending:

It generates a relation between policy and catalogue. It defines which catalogue falls under which category and thus defines its duration of issue.

13. Penalty:

This generates information from the issue table and records for return date along with category name and hence has definite return time for all types of resources. Hence, puts a penalty if delayed according to date and time.

14. Policy:

This table is highly useful as it generates return time for resources according to its category type. It stores the duration of issue according to the category of a particular catalogue.

15. Recommendation:

Allows professors to add course recommendation books in the website to help the library keep its availability on a regular basis and timings.

16. Return :

This table stores the return date for a particular book which is issued following a remark , not necessary. This takes issue id from the issue table and generates return date according to the lending policy of a particular catalogue which is defined in the policy table.

17. Rooms :

This table stores the information about the availability of the rooms and its entry and exit time.

CONTRIBUTIONS:

1. Aashmun Gupta -

Worked on SQL Injection Attacks and fixes.

2. Anmol Kumar -

Worked on XSS Attacks, Brute Force Attacks, and Google Login.

3. Anushri Sanodia -

Worked on Multi-User Access Control.

4. Deepanjali Kumari -

Worked on database changes suggested by stakeholders.

5. Dhruv Sharma -

Worked on database changes

6. Pavani Khale -

Worked on Database Locks and feedback from stakeholders.

7. Yash Patel -

Worked on SQL Injection

References:

1. Flask App Login with Google -

<https://dev.to/marlanna/flask-app-login-with-google-3j24>