

Indian Institute of Technology Gandhinagar



Library Management System

CS 432 : Project Report

Group - X

Team Members

Jayesh Bhadange	20110082
Pranav Rathod	20110143
Naval Jaggi	20110118
Harendra Khatik	20110072
Sparsh Dawra	20110203
Manpreet Singh	20110109
Sidharth Joshi	19110169
Gajanan Donge	20110061
Sankarshan	20110184
Riya Dhantoliya	20110168

CS 432: Databases

Assignment 4: DEPLOYING THE DBMS

3. Tasks

3.1 Responsibility of G1:

1)

1. Before first feedback -

The navigation bar of the admin view for website included the Home page which had the search bar located at the bottom of the web page, Books page included the insertion, deletion and update options for the new or issued entries of the book, Author section had the author corresponding to the books present under the books section.

2. After first feedback -

Stakeholders found our website user friendly and useful. In addition to this, they focussed on the dropdown feature to be implemented for surfing through various books as there might be redundancy based on books and authors as multiple authors might have multiple books hence searching feature would take time for the particular books to get searched. They also asked to add any more advanced feature than the existing library catalog website at IIT Gandhinagar.

3. After second feedback -

After adding the dropdown feature in the search bar by filtering the searches according to books or authors and creating the user view for the existing users of the library catalog that included books issued. Stakeholders asked to put up a selection feature that had more advanced features like boolean which is used when millions of data are supposed to be inserted and also that keyword used in search feature should have been optimized. _____(New changes made in the website as desired)

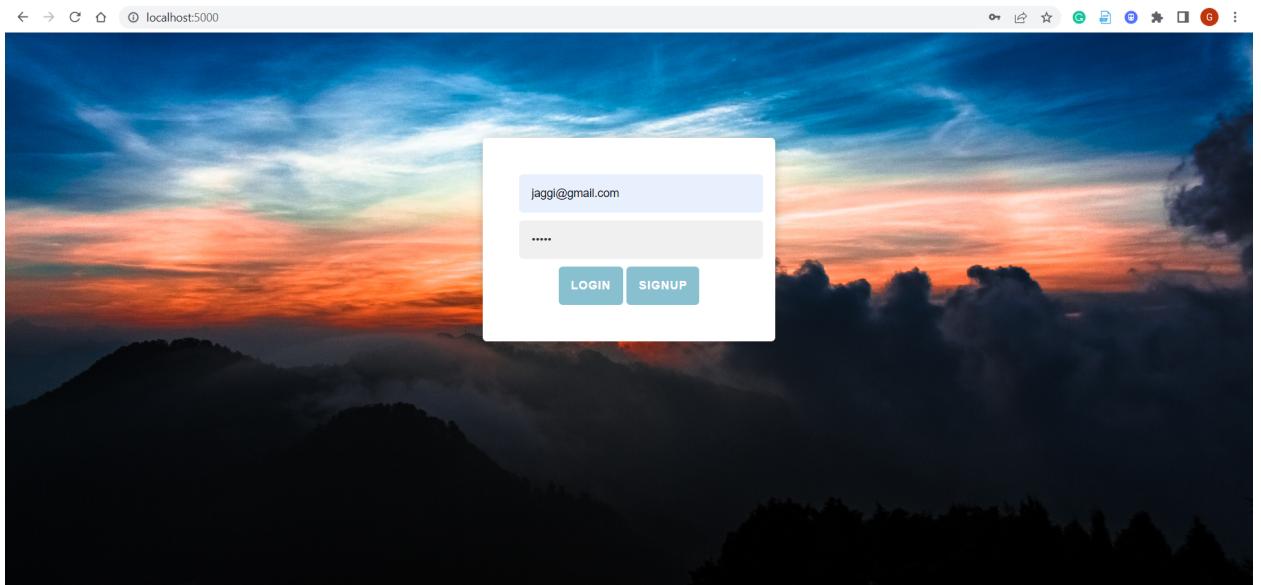
2). There are two classes of user for library-management

a) Admin/ stakeholders

For admin login, there are different user credentials.

Email ID - **Jaggi@gmail.com**

Password - **harry**



After login, the following page will appear for admin, where he can update book, authors, publishers information. Also he can view users' information but not be able to view the password of users, it will be available only for database owners.

A screenshot of a web application titled 'Library Catalog'. The header includes links for Home, Books, Authors, publisher, shelf, Users, and Logout. Below the header is a photograph of a library interior with a blue globe and bookshelves. To the right of the image is a quote: "The only thing that".

On clicking the book navigation bar admin can access the book information shown below.

Here the admin can insert new books and delete old books.

Library Management System

Book_ID	Title	Edition	Copies	Availability	Author_ID
1	The Great Gatsby	1	10	1	1
2	To Kill a Mockingbird	2	15	1	2
3	1984	1	5	0	1
4	Pride and Prejudice	3	20	1	3
5	The Catcher in the Rye	2	8	0	4
6	The Hobbit	1	12	1	5
7	Lord of the Flies	1	6	1	3
8	Animal Farm	2	10	0	8
9	The Odyssey	1	7	1	7
10	The Sun Also Rises	3	3	1	2
11	The Princeton Companion to Mathematics	1	10	1	1
12	Introduction to Algorithms	3	8	1	2
13	The Elements of Statistical Learning: Data Mining, Inference, and Prediction	2	12	1	3
14	A Mathematician's Apology	1	5	1	4
15	How to Solve It	2	7	1	5
16	The Feynman Lectures on Physics	1	10	1	6
17	University Physics with Modern Physics	14	8	1	7
18	Introduction to Electrodynamics	4	12	1	7
19	Gravitation	1	5	1	9
20	Quantum Mechanics: Concepts and Applications	2	7	1	10
21	Chemical Principles: The Quest for Insight	3	10	1	1
22	Organic Chemistry	2	8	1	2
23	Principles of Modern Chemistry	8	12	1	3
24	Physical Chemistry: Thermodynamics, Structure, and Change	10	5	1	4
25	Biochemistry	7	7	1	5

[INSERT](#)[DELETE](#)

On the author page also the admin is able to update information.

Library Management System

author_ID	author_Name
1	J K Rowling
2	Stephen King
3	Margaret Atwood
4	James Baldwin
5	Jane Austen
6	Gabriel Garcia Marquez
7	Chimamanda Ngozi Adichie
8	Toni Morrison
9	Haruki Murakami
10	Salman Rushdie

[INSERT](#)[DELETE](#)

On the publisher page admin can update information.

Library Management System

publisher id	publisher Nmae
11	Penguin Random House
12	HarperCollins
13	Simon & Schuster
14	Hachette Livre
15	Macmillan Publishers
16	Scholastic Corporation
17	Bloomsbury Publishing
18	Pearson Education
19	Oxford University Press
20	Cambridge University Press

Also there will be a shelf assigned to books of the same subjects to which the location is also accessible for admin such that he can change it accordingly.

Library Management System

section id	Row no.	Department
1	1	Novels
1	2	Fiction
2	1	Math
2	2	Physics
3	1	chemistry

Admin can access the user information and can update it according to users request.

① localhost:5000/libuser

Library Management System

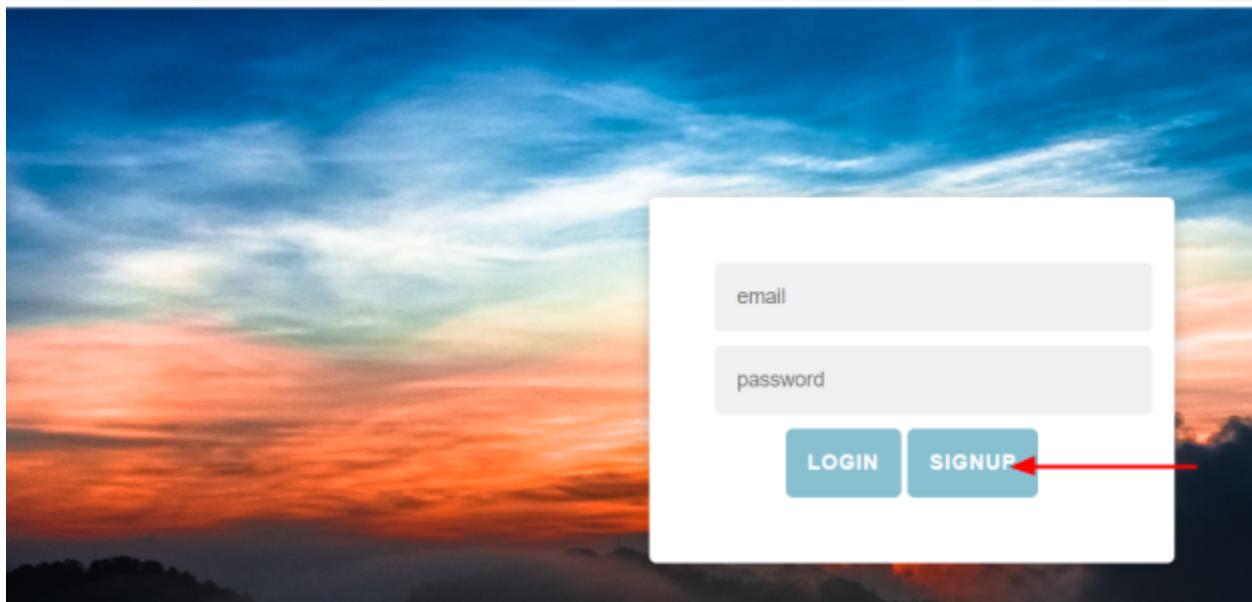
UserID	first_name	middle_name	end_name	Contact
1	John	David	Doe	john.doe@example.com
2	Jane	Elizabeth	Smith	jane.smith@example.com
3	Michael	Patrick	Scott	michael.scott@example.com
4	Pamela	Morgan	Halpert	pamela.halpert@example.com
5	Jim	Duncan	Halpert	jim.halpert@example.com
6	Angela	Noelle	Martin	angela.martin@example.com
7	Kevin	Jay	Malone	kevin.malone@example.com
8	Oscar	Gutierrez	Martinez	oscar.martinez@example.com
9	Toby	Wyatt	Flenderson	toby.flenderson@example.com
10	Stanley	James	Hudson	stanley.hudson@example.com

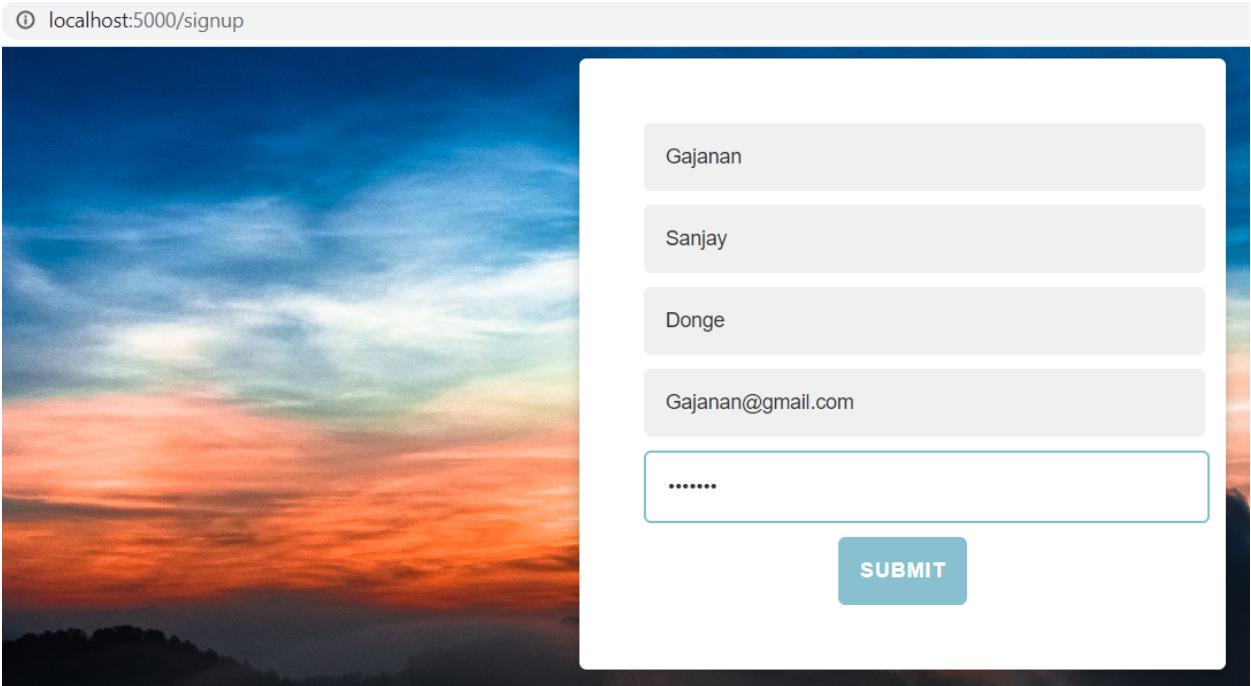
Update Contact DELETE RENAME

b) Library users (students, faculty)

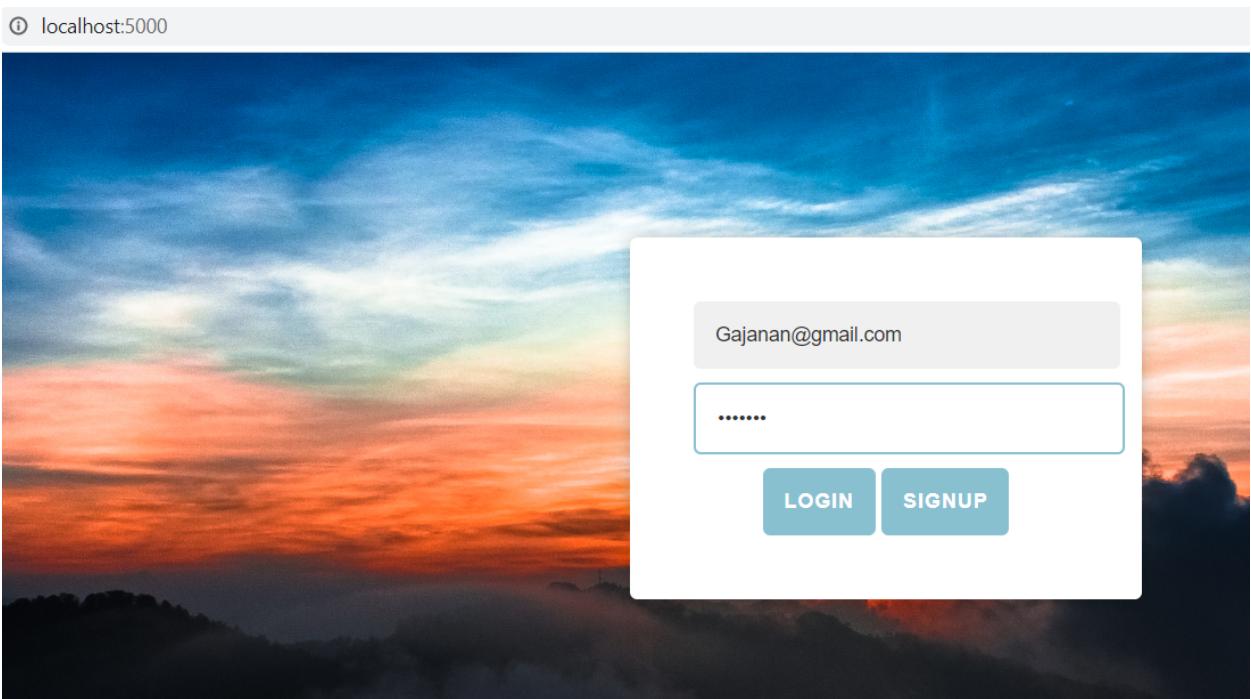
If the user is not registered with the library then he can sign up.

① localhost:5000





So user information will be added into the database. Now user can login.



The main page will be open for user.

localhost:5000/dbmsuser

Here user can see which books are available, authors,publisher and can issue book.

When user will click any navigation bar, he will only able to read table.

localhost:5000/books

Library Management System

Book_ID	Title	Edition	Copies	Availability	Author_ID
1	The Great Gatsby	1	10	1	1
2	To Kill a Mockingbird	2	15	1	2
3	1984	1	5	0	1
4	Pride and Prejudice	3	20	1	3
5	The Catcher in the Rye	2	8	0	4
6	The Hobbit	1	12	1	5
7	Lord of the Flies	1	6	1	3
8	Animal Farm	2	10	0	8
9	The Odyssey	1	7	1	7
10	The Sun Also Rises	3	3	1	2
11	The Princeton Companion to Mathematics	1	10	1	1
12	Introduction to Algorithms	3	8	1	2
13	The Elements of Statistical Learning: Data Mining, Inference, and Prediction	2	12	1	3
14	A Mathematician's Apology	1	5	1	4
15	How to Solve It	2	7	1	5
16	The Feynman Lectures on Physics	1	10	1	6
17	University Physics with Modern Physics	14	8	1	7
18	Introduction to Electrodynamics	4	12	1	7
19	Gravitation	1	5	1	9
20	Quantum Mechanics: Concepts and Applications	2	7	1	10
21	Chemical Principles: The Quest for Insight	3	10	1	1
22	Organic Chemistry	2	8	1	2
23	Principles of Modern Chemistry	8	12	1	3
24	Physical Chemistry: Thermodynamics, Structure, and Change	10	5	1	4
25	Biochemistry	7	7	1	5

① localhost:5000/authors

Library Management System

author_ID	author_Name
1	J K Rowling
2	Stephen King
3	Margaret Atwood
4	James Baldwin
5	Jane Austen
6	Gabriel Garcia Marquez
7	Chimamanda Ngozi Adichie
8	Toni Morrison
9	Haruki Murakami
10	Salman Rushdie

① localhost:5000/publisher

Library Management System

publisher id	publisher Name
11	Penguin Random House
12	HarperCollins
13	Simon & Schuster
14	Hachette Livre
15	Macmillan Publishers
16	Scholastic Corporation
17	Bloomsbury Publishing
18	Pearson Education
19	Oxford University Press
20	Cambridge University Press

① localhost:5000/shelf

Library Management System

section id	Row no.	Department
1	1	Novels
1	2	Fiction
2	1	Math
2	2	Physics
3	1	chemistry

If a user wants to issue a book but does not know about book information he can search the book at the search bar added to the bottom of the webpage.

Search Books

Search By: Search Query:

© 2023 Library Catalog

Search Books

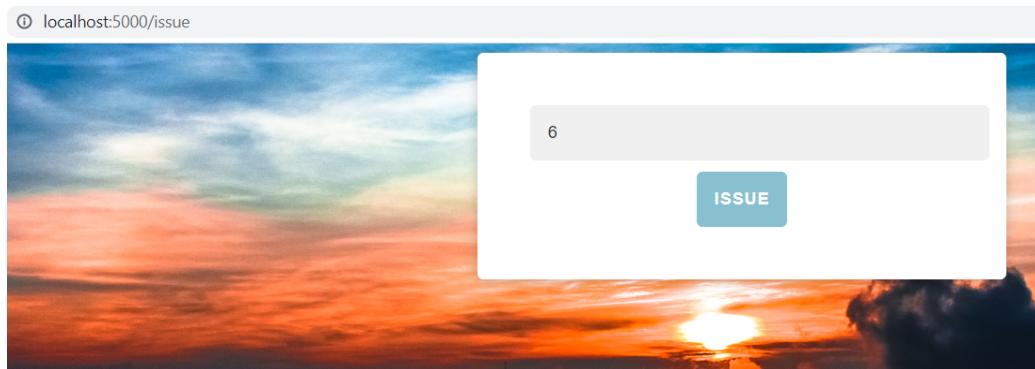
Search By: Search Query:

← → C ⌂ ① localhost:5000/dbmsuser

Library Management System

Book_ID	Title	Edition	Copies	Availability	Author_ID	Section_ID	Row_No
6	The Hobbit	1	12	1	5	1	2

So here the user know id of the book so he can issue book.



After that a database table will be updated with user email id, book id and issue date with time.

	issue_id	book_id	email	issue_date
▶	1	5	john.doe@example.com	2023-04-15 02:56:44
	2	2	john.doe@example.com	2023-04-15 10:50:55
	3	6	Gajanan@gmail.com	2023-04-15 23:48:26
●	NULL	NULL	NULL	NULL

3.2 Responsibility of G2:

1 Lock Implementation

The below images show the snippet of code where the locks are applied. Locks are essential to maintain consistency in the database. Without locks, concurrent transactions could modify the same data simultaneously, leading to inconsistent or incorrect results.

```
github.com/jayeshbadange/Assignment4_GroupX/blob/main/GroupX/app.py

def delete():
    if request.method == 'POST':
        userDetails = request.form
        book_id = userDetails['book_id']

        cur = mysql.connection.cursor()
        cur.execute("LOCK TABLES book WRITE") # Lock the table you want to query
        cur.execute("DELETE FROM book WHERE book_id = %s", [book_id])

        mysql.connection.commit()
        cur.execute("UNLOCK TABLES")# Unlock the table
        cur.close()
        return redirect("/DBMS")

    return render_template('delete.html')

@app.route('/updateuser', methods=['POST', 'GET'])
def update_user():
    if request.method == 'POST':
        userDetails = request.form
        UserID = userDetails['UserID']
        Contact = userDetails['Contact']
        cur = mysql.connection.cursor()
        cur.execute("SELECT * FROM users WHERE UserID = %s", [UserID])
        user = cur.fetchone()

        if user:
            cur.execute("LOCK TABLES users WRITE") # Lock the table you want to query
            cur.execute("UPDATE users SET Contact = %s WHERE UserID = %s", [Contact, UserID])

            mysql.connection.commit()
            cur.execute("UNLOCK TABLES")# Unlock the table
            cur.close()
            flash('User updated successfully!', 'success')

    return render_template('updateuser.html')
```

```
 187 @app.route('/issue', methods=['GET', 'POST'])
 188 def issue():
 189     if request.method == 'POST':
 190         # Retrieve the user input from the form
 191         book_id= request.form['book_id']
 192         # Retrieve the user's email from the session
 193         email = session.get('email')
 194
 195         # Open a database connection
 196         cur = mysql.connection.cursor()
 197         # Check if the book available or not
 198         cur.execute("SELECT * FROM book WHERE book_id = %s AND copies > 0", [book_id])
 199         b = cur.fetchone()
 200
 201         if b is None:
 202             flash('Book is not available.', 'error')
 203             return redirect('/issue')
 204         # Get the current date and time and format it as a string
 205         issue_date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
 206         cur.execute("LOCK TABLES book_issue WRITE") # Lock the table you want to query
 207         cur.execute("INSERT INTO book_issue (book_id, email,issue_date) VALUES (%s, %s,%s)", (book_id, email,issue_date))
 208         mysql.connection.commit()
 209         cur.execute("UNLOCK TABLES")# Unlock the table
 210         # Update the available copies in the book table
 211         cur.execute("LOCK TABLES book WRITE") # Lock the table you want to query
 212         cur.execute("UPDATE book SET copies = copies - 1 WHERE book_id = %s", (book_id,))
 213         mysql.connection.commit()
 214         cur.execute("UNLOCK TABLES")# Unlock the table
 215         flash('Book issued successfully.', 'success')
 216         sleep(2)
 217         return redirect('/dbmsuser')
 218     return render_template('issue.html')
 219
 220 @app.route('/Issue', methods=['GET', 'POST'])
 221
```

```
235     if request.method == 'POST':
236         userDetails = request.form
237         old_col_name = userDetails['old_col_name']
238         new_col_name = userDetails['new_col_name']
239         cur = mysql.connection.cursor()
240         cur.execute("ALTER TABLE `users` RENAME COLUMN `old_col_name` TO `new_col_name`;", [])
241         mysql.connection.commit()
242         cur.close()
243         render_template('libuser.html')
244         return redirect('/libuser')
245     return render_template('rename.html')
246
247
248 @app.route('/add', methods=['GET', 'POST'])
249 def add():
250     if request.method == 'POST':
251         userDetails = request.form
252         book_id = userDetails['book_id']
253         title = userDetails['title']
254         edition = userDetails['edition']
255         copies = userDetails['copies']
256         Availability_status = userDetails['Availability_Status']
257         author_id = userDetails['author_id']
258         cur = mysql.connection.cursor()
259         cur.execute("LOCK TABLES book WRITE") # Lock the table you want to query
260         cur.execute("INSERT INTO book(book_id, title, edition, copies, Availability_Status, author_id) VALUES(%s,%s,%s,%s,%s,%s)", (book_id, title, edition, copies, Availability_status, author_id))
261         mysql.connection.commit()
262         cur.execute("UNLOCK TABLES")# Unlock the table
263         cur.close()
264         return redirect('/DBMS')
265     return render_template('add.html')
266
267 @app.route('/delete', methods=['POST', 'GET'])
268 def delete():
269     return
```



2 Changes as per feedback from library stakeholders.

Library stakeholders gave feedback to us that sometimes user know only some keyword of title of book or author id and book id so create a dropdown system where search can be made using all options.

Search Books

Search By: Search Query:

© 2023 Library Management System

Search Books

Search By: Search Query:

localhost:5000/dbmsuser

Library Management System

Book_ID	Title	Edition	Copies	Availability	Author_ID	Section_ID	Row_No
6	The Hobbit	1	11	1	5	1	2

Search Books

Search By: Search Query:

localhost:5000/dbmsuser

Library Management System

Book_ID	Title	Edition	Copies	Availability	Author_ID	Section_ID	Row_No
5	The Catcher in the Rye	2	8	0	4	1	1

Search Books

Search By: Search Query:

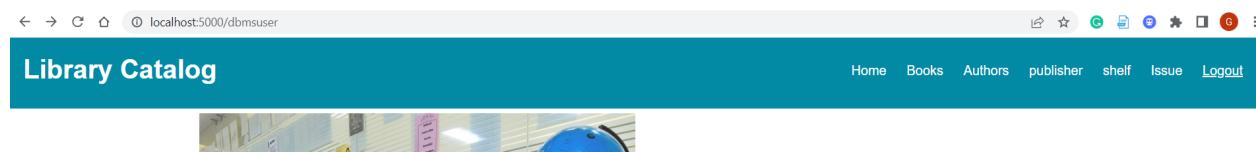
Book_ID	Title	Edition	Copies	Availability	Author_ID	Section_ID	Row_No
2	To Kill a Mockingbird	2	15	1	2	1	1
10	The Sun Also Rises	3	3	1	2	1	2
12	Introduction to Algorithms	3	8	1	2	2	1
22	Organic Chemistry	2	8	1	2	3	1

Also library members suggested we create a logout option because user security is also important.

Before feedback

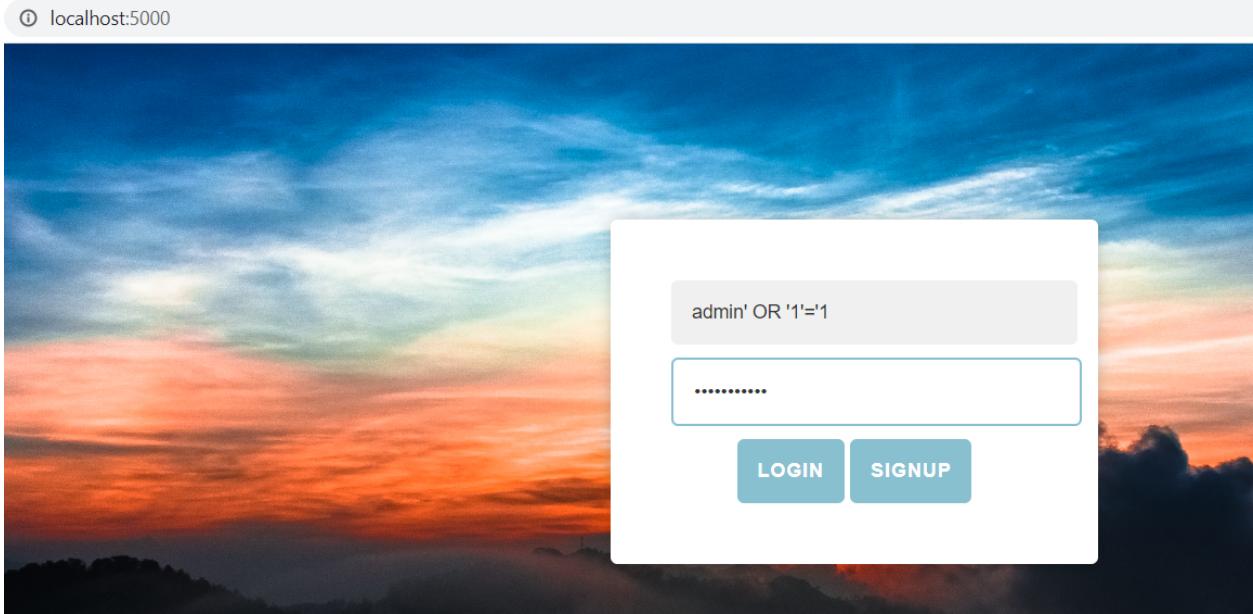


After feedback



3.3 Responsibility of G1 and G2:

1.

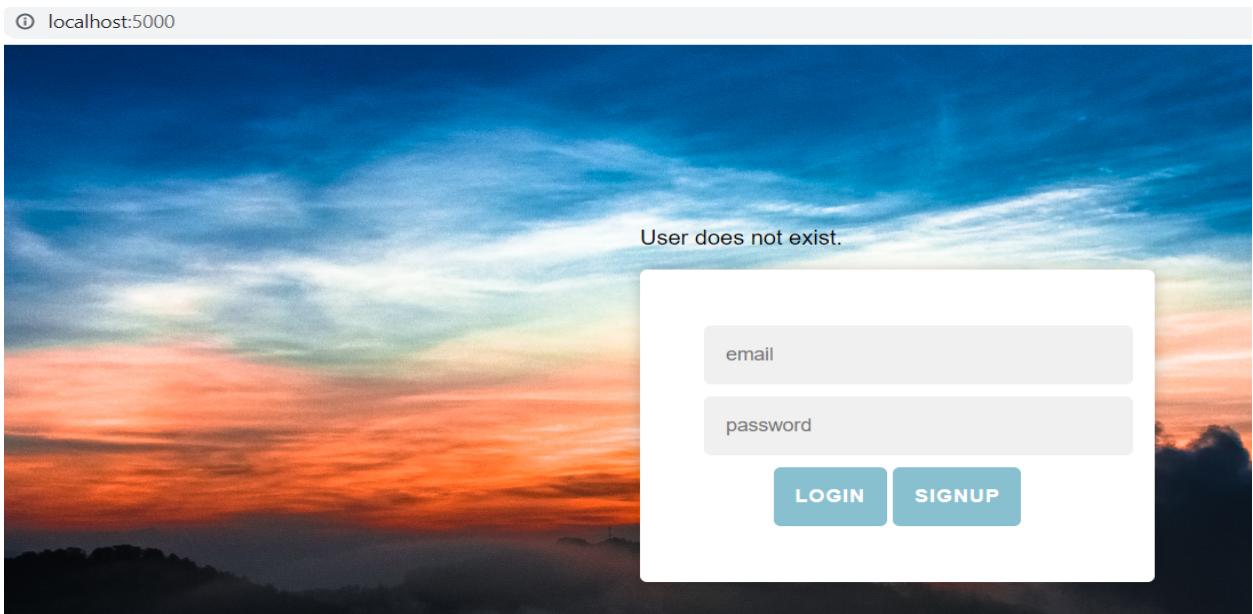


Username: admin' OR '1'='1

Password: password123

It will run SQL Query:

SELECT * FROM user WHERE email = '**admin' OR '1'='1'** AND password ='**password123**'



So our website defends from SQL injection.

- 2. Showing 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.**

Entities and Attributes:-

<i>Serial No</i>	<i>Entity Name</i>	<i>Attribute</i>
1	Book	BookID(Primary Key), Author, Title, Edition, Copies, AvailabilityStatus
2	Publisher	Name, PublisherID(Primary Key)
3	Issue	IssueID(Primary Key) IssueDate, DueDate
4	Return	ReturnDate, ReturnID(Primary Key)
5	Fine	FineID(Primary Key), DueAmount, PayDate
6	User	UserID(Primary Key), Name, Contact
7	Student	Program, MaxBooks
8	Faculty	Dept, Office
9	Purchase	PurchaseID(Primary Key), Price, PurchaseDate
10	Shelf	RowNo(Primary Key), SectionID(Primary Key), SectionDept

We have done work on and represented most of the relations which we got after second feedback. You can see these in the navigation bar itself. We made a different webpage for each relation so that they can process their query without any confusion and relevant features of UPDATE, DELETE, RENAME, etc are available corresponding to the feature.

Team- G1: feedback from library and document

- 1) Sankarshan Kulkarni (20110184)
- 2) Riya Dhantoliya (20110168)
- 3) Gajanan Donge (20110061)
- 4) Siddharth Joshi (19110169)

Team - G2: Updated website according to feedback, locks , injection.

- 1) Jayesh Bhadange (20110082)
- 2) Pranav Rathod (20110143)
- 3) Naval Jaggi (20110118)
- 4)Gajanan Donge(20110061)
- 5) Harendra Khatik (20110072)
- 6) Sparsh Dawra (20110203)
- 7) Manpreet Singh (20110109)