CS 6314.002 Web Programming Languages

Dallas Public Library

Team: 4
Harshita Rastogi (HXR190001)
Prachi Vats (PXV180021)
Shivam Gupta (SXG190040)

CONTENTS

PROJECT DESCRIPTION

Our project is an implementation of a public library where customers are able to view a list of books, each having multiple copies and are able to issue the books for a week. The Library lets you borrow multiple books at once, however there's a minimum fine of $0.75/each day if the book is returned after the due date.

Important Components of the System

The main important actors in the domain are - Admin and Customers, which form the very basic foundation of the Public Library.

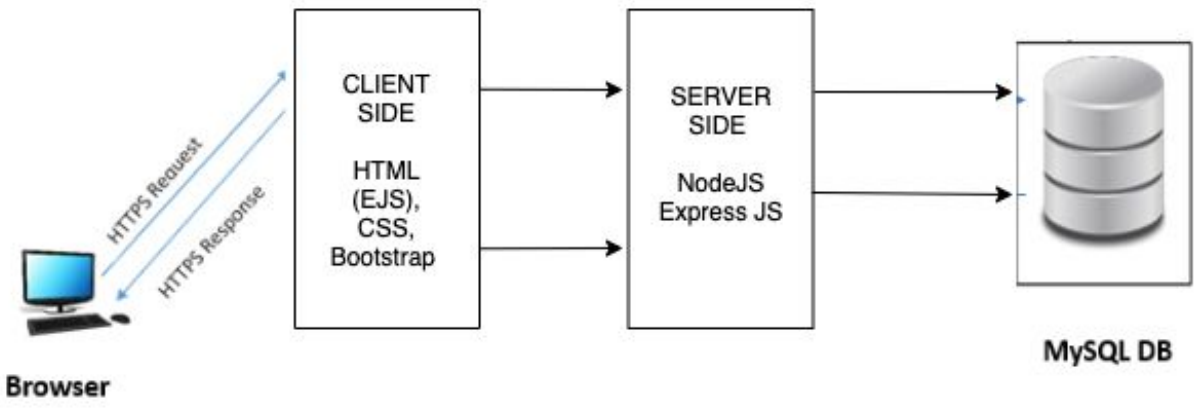
- **Admin** - a person who has additional access to features of the public library portal who is able to view all the books in the library (issued and currently available), add new books, delete books, edit books, add customers, remove customers.

  Only the admin can add/change the picture of the book added in the library portal.

- **Customer** - a person who is registered on to the public library portal and has the ability to view books, issue books and return books.

ARCHITECTURE

The Online Public Library is a full fledged scalable web based application based on NodeJS, Express, EJS (Embedded Javascript templating), CSS (Cascading Style Sheets), Bootstrap and MySQL for the database. The architecture of the entire web application is as follows:



TECHNOLOGIES USED

➔ **FRONT END DESIGN**
For designing the front end on EJS files, CSS and bootstrap components have been used.

➔ **BACK-END DESIGN**
For server side scripting, we have used NodeJS and Express JS.
A relational database has been made for the application which uses MySQL.

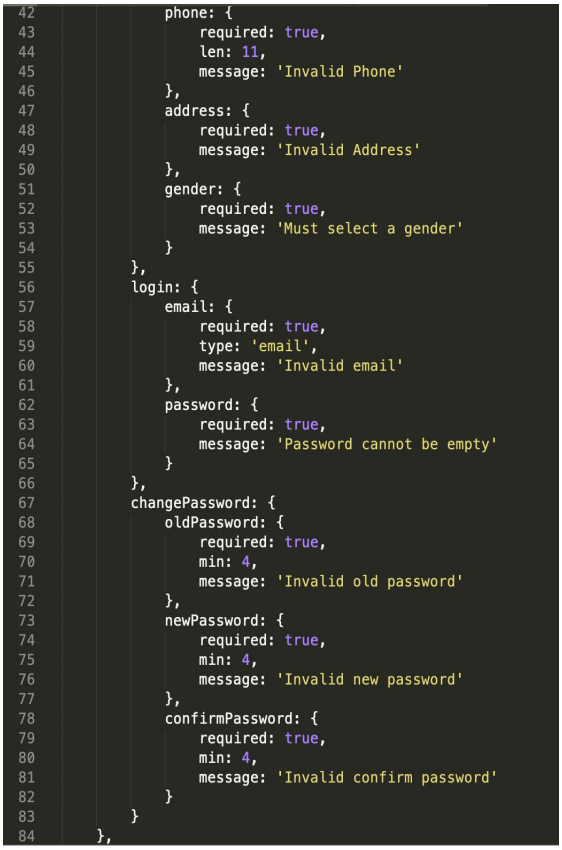
FUNCTIONALITIES

1. **HOMEPAGE**
   The main page shows the user an option to either create a new account into the public library system or login using the registered credentials.

2. **USER SIGNUP**
   On clicking the signup button on the homepage, it redirects to '/signup' where the user needs to entire the following details:

   - Validation Rules are specified in validation_rules >> **rules.js**.

```
    users: {
        create: {
            name: {
                required: true,
                message: 'Name cannot be empty'
            },
            email: {
                required: true,
                type: 'email',
                message: 'Invalid email'
            },
            phone: {
                required: true,
                len: 11,
                message: 'Invalid Phone'
            },
            password: {
                required: true,
                min: 4,
                message: 'Invalid Password'
            },
            address: {
                required: true,
                message: 'Invalid Address'
            },
            gender: {
                required: true,
                message: 'Must select a gender'
            }
        },
        update: {
            name: {
                required: true,
                message: 'Name cannot be empty'
            },
            email: {
                required: true,
                type: 'email',
                message: 'Invalid email'
            },
            phone: {
                required: true,
                len: 11,
```



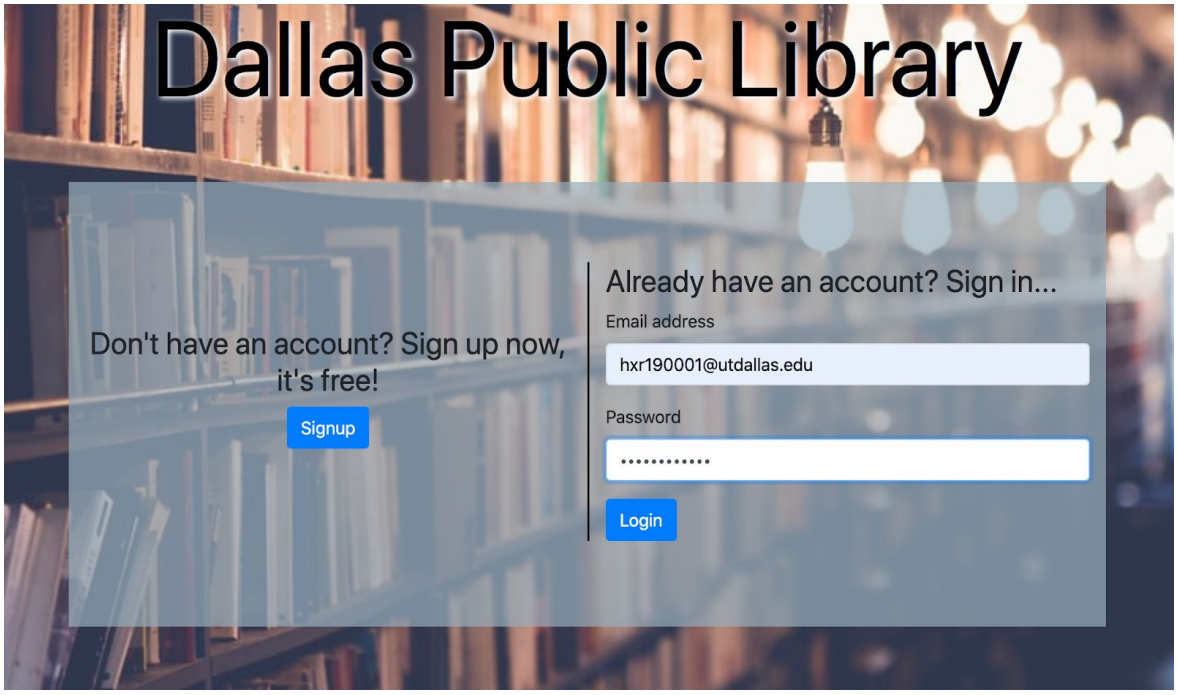
```
                len: 11,
                message: 'Invalid Phone'
            },
            address: {
                required: true,
                message: 'Invalid Address'
            },
            gender: {
                required: true,
                message: 'Must select a gender'
            }
        },
        login: {
            email: {
                required: true,
                type: 'email',
                message: 'Invalid email'
            },
            password: {
                required: true,
                message: 'Password cannot be empty'
            }
        },
        changePassword: {
            oldPassword: {
                required: true,
                min: 4,
                message: 'Invalid old password'
            },
            newPassword: {
                required: true,
                min: 4,
                message: 'Invalid new password'
            },
            confirmPassword: {
                required: true,
                min: 4,
                message: 'Invalid confirm password'
            }
        }
    },
```

3. **EXISTING USER LOGIN**

   On the homepage, the existing user can enter the registered email address and password to login to the portal. One can login on the portal either as an admin or a user.

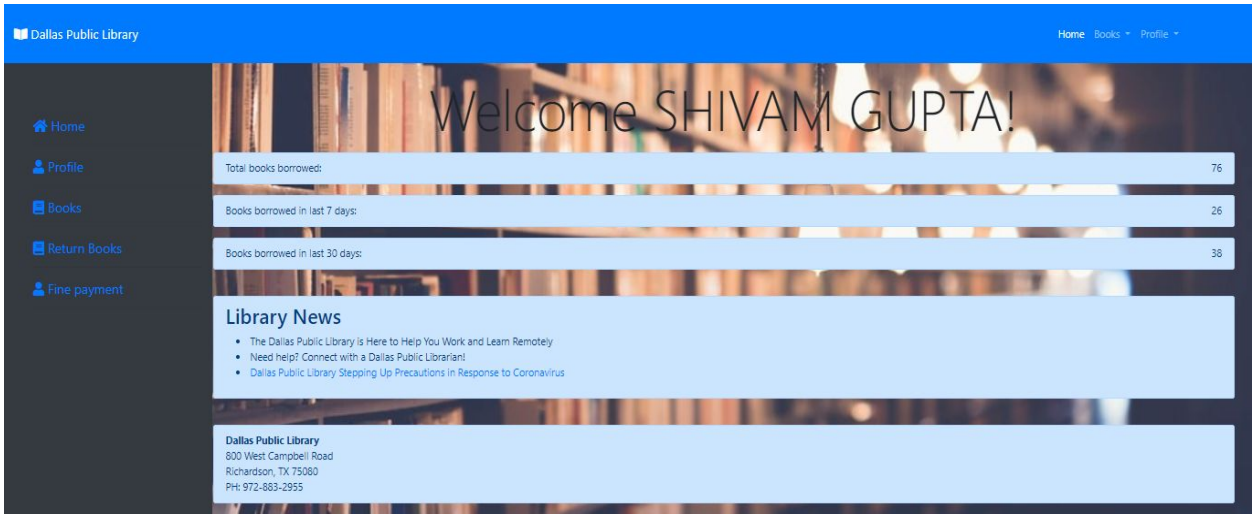
   On clicking on the Login button, the form first checks the database for the right credentials entered. If successful, the user will be logged in and taken to the home page. If not, then the ‘Invalid Email or Password’ message will be displayed.

   Email address - *REQUIRED
   Password - *REQUIRED



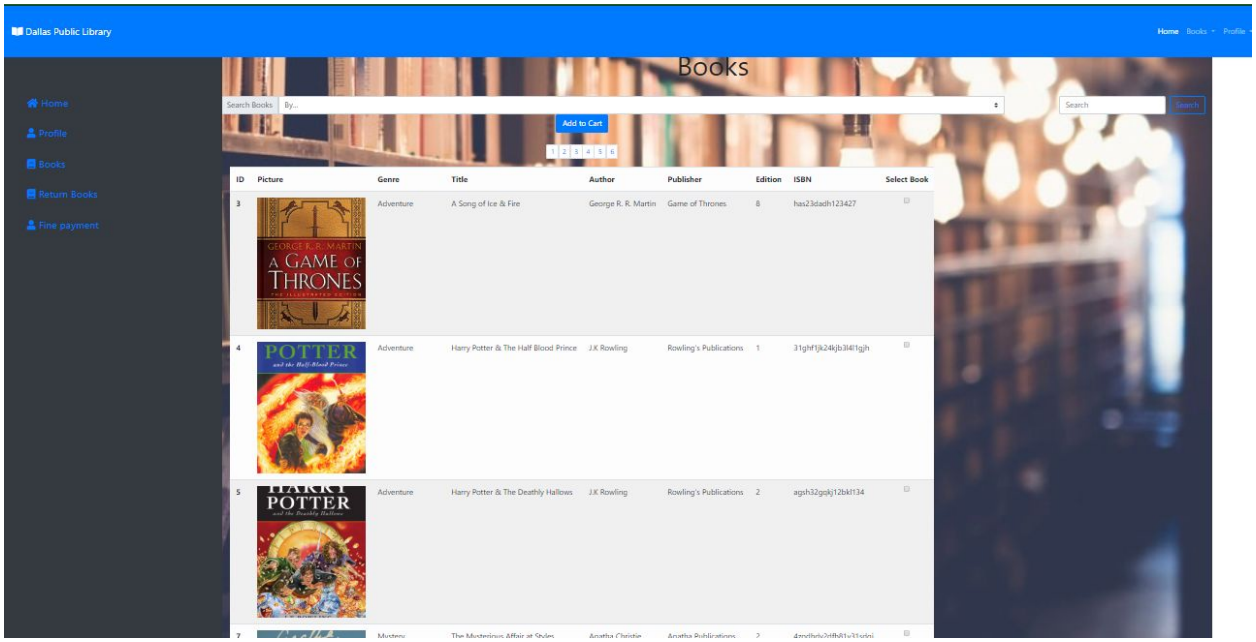
4. CUSTOMER HOMEPAGE

Customer can view total the total borrowed books, books borrowed in last 7 days, Library News and there is a navbar for navigation to other pages.



5. BOOKS

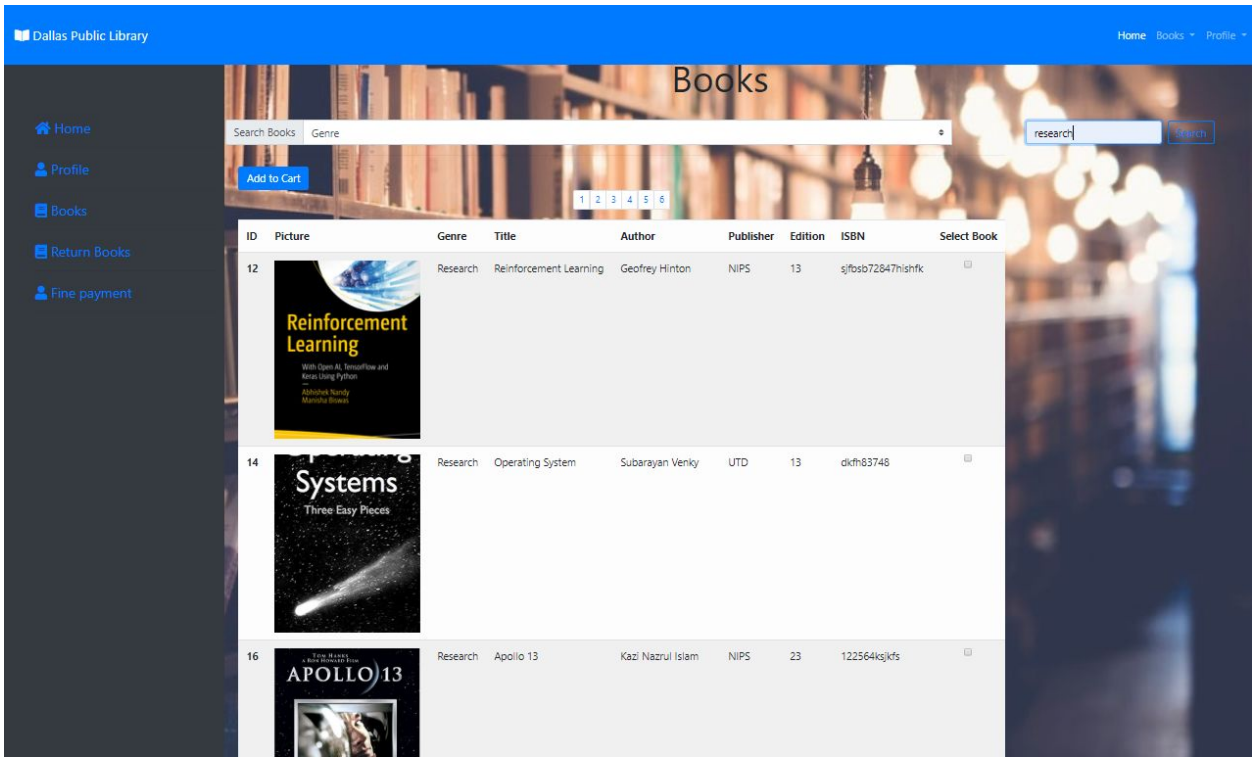
All the books can be viewed on the books page and the user can filter and search for the books as per the criteria such as Title, Genre, Author, Publisher etc. And even add books to cart.



6. "SEARCH" FOR A SPECIFIC BOOK

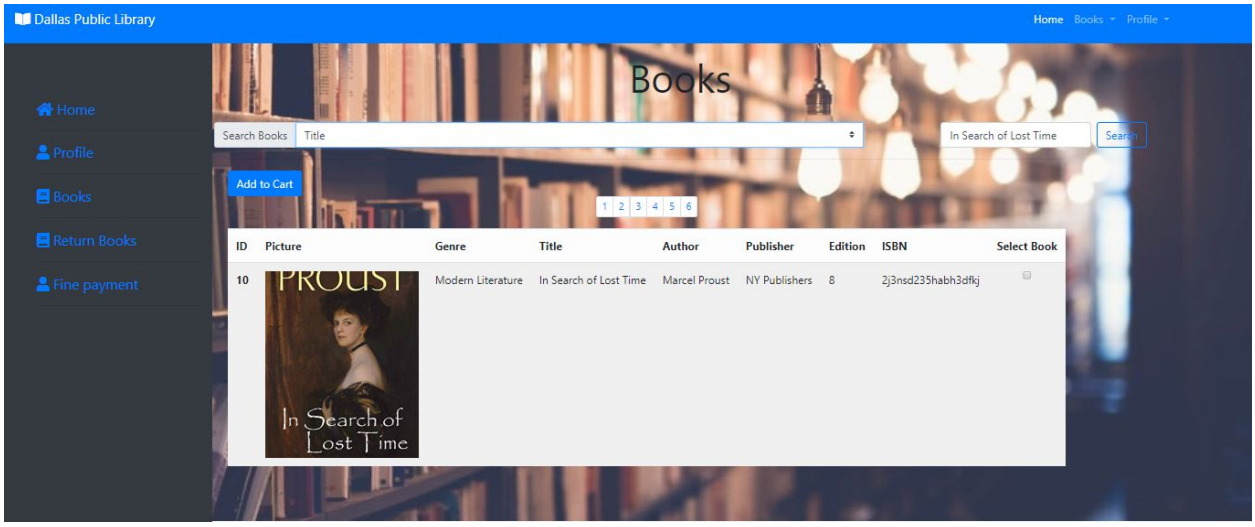
CASE 1:

If we search a book by a filter "Genre" and enter Genre as research then it shows all the books of research. You can also search by other attributes like Titles, author. etc .



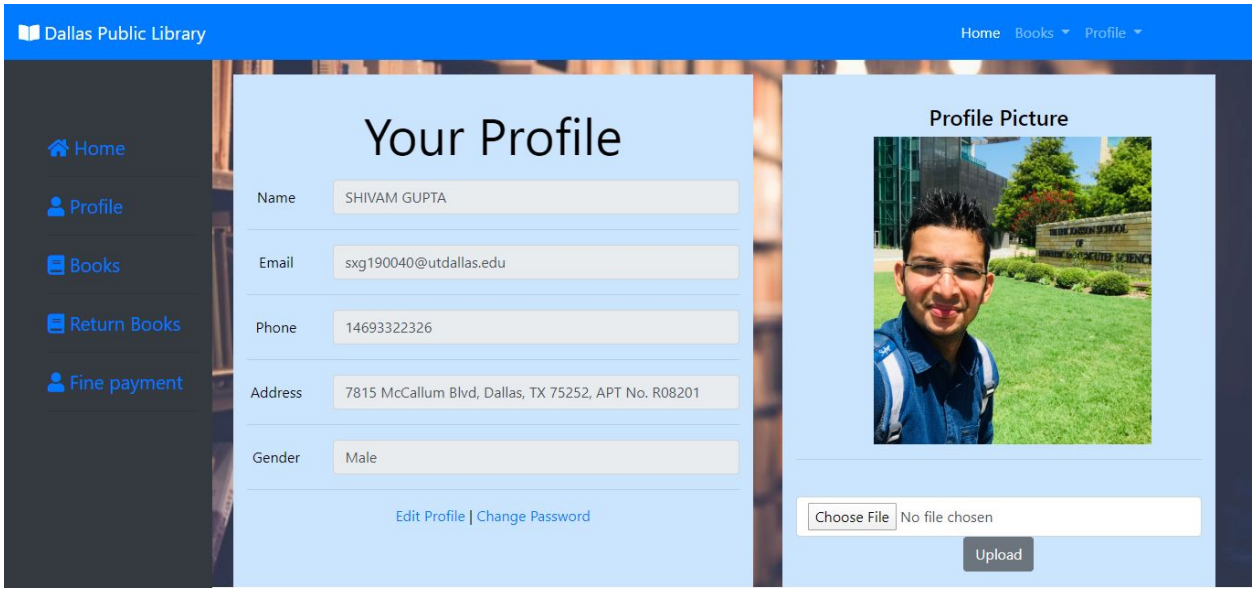
CASE 2:

Filtering based on genre “title” In Search of Lost Time



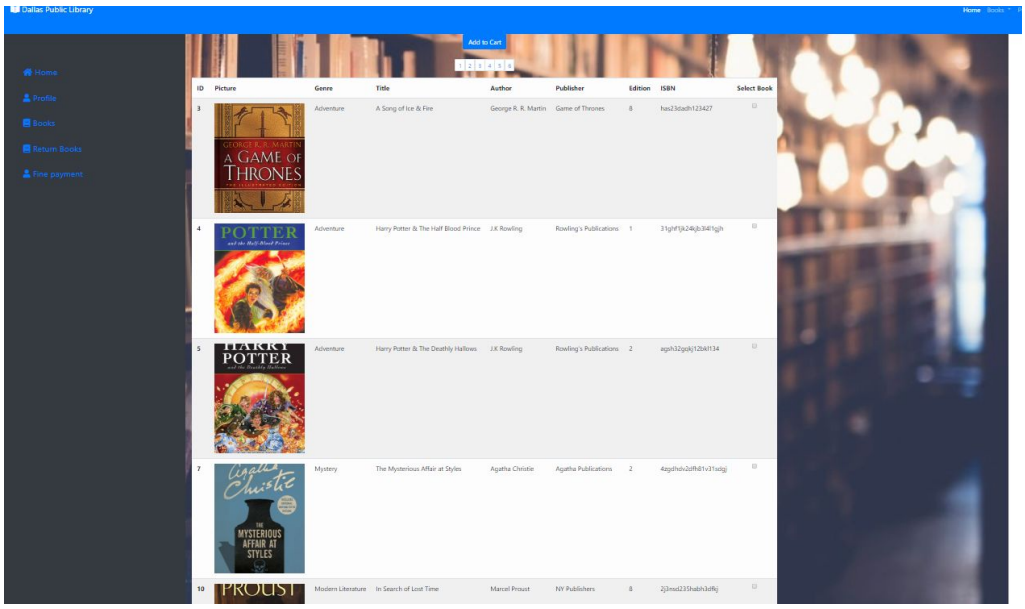
7. CUSTOMER PROFILE

Customer can edit the profile i.e Name, Email, Phone, address, Gender and can even change the profile picture.



8. PAGING FUNCTIONALITY

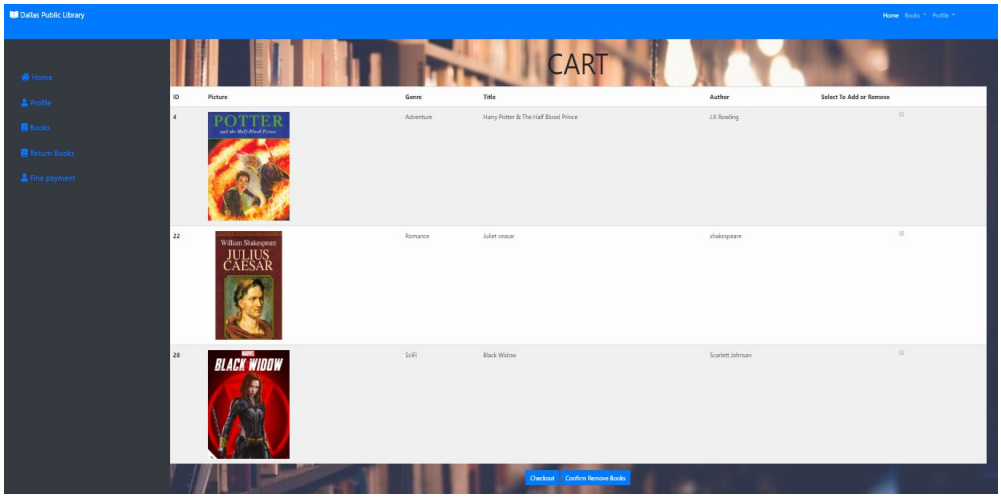
Paging functionality includes dividing the images into several pages like when a user searches, the data fetched is huge, so we need to divide the pages into different segments.
For Example: Each page shows 5 books with their details.



9. CART

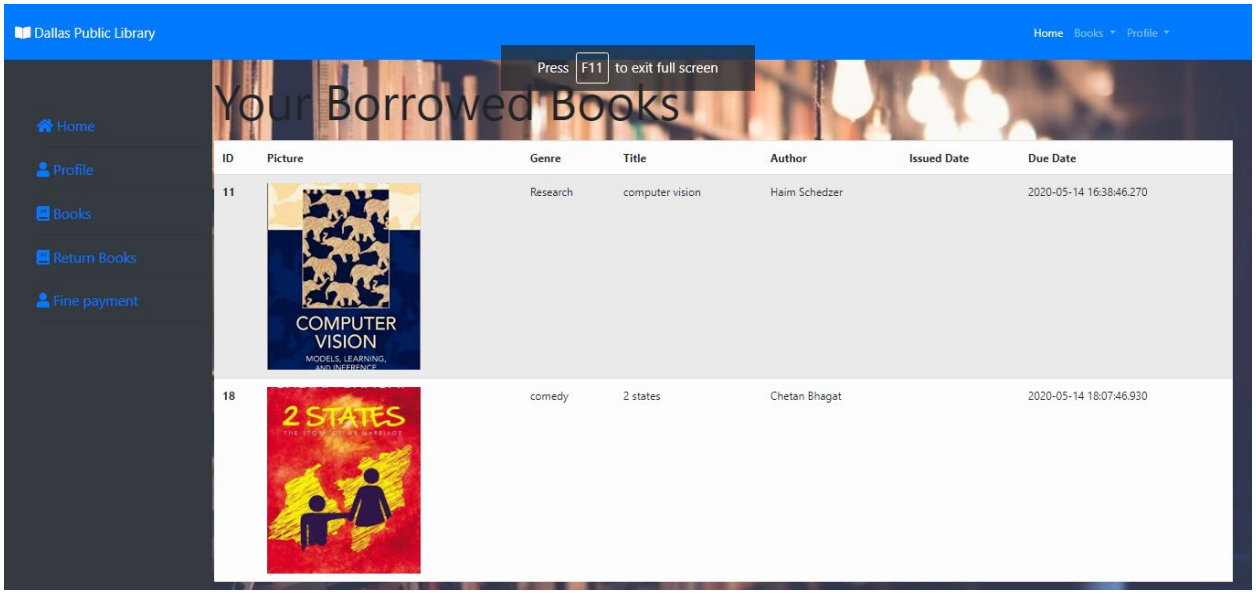
After choosing the particular books to borrow, the books go into the cart. From the cart, the user has the option of finally choosing the books to borrow.

To borrow, the user needs to click on the “Checkout” button to borrow it.



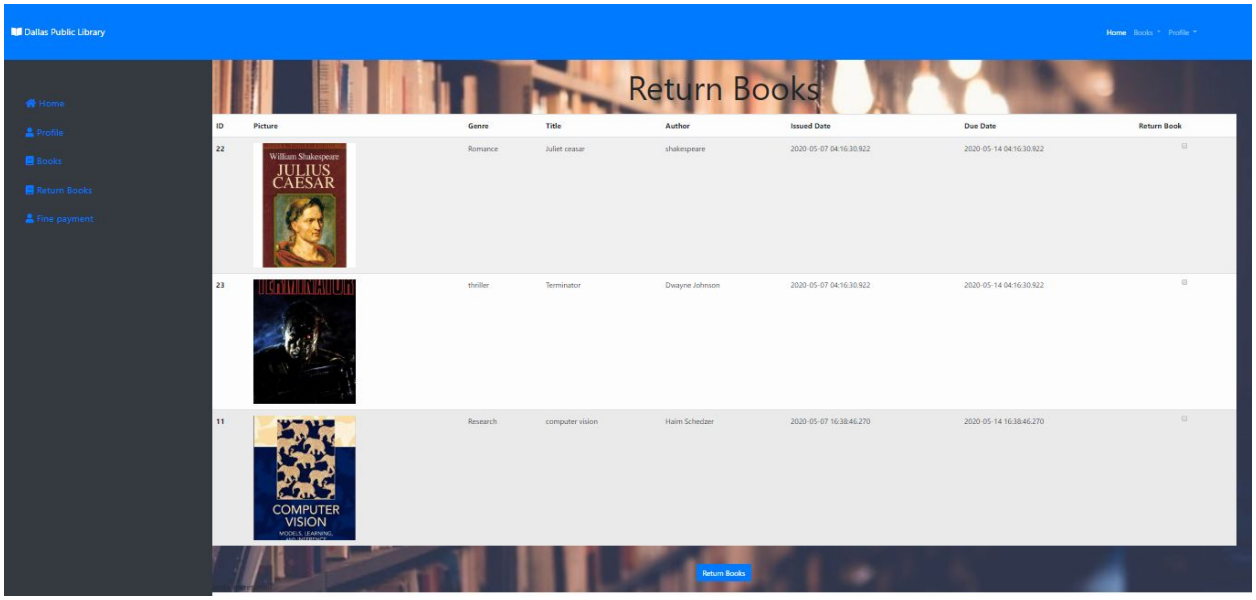
10. VIEW BORROWED BOOKS

Once the books are checked out, the currently borrowed books can be viewed on the ‘You Borrowed Books’ page. The page contains details regarding the books borrowed, time of borrow and the due date to return the issued book.



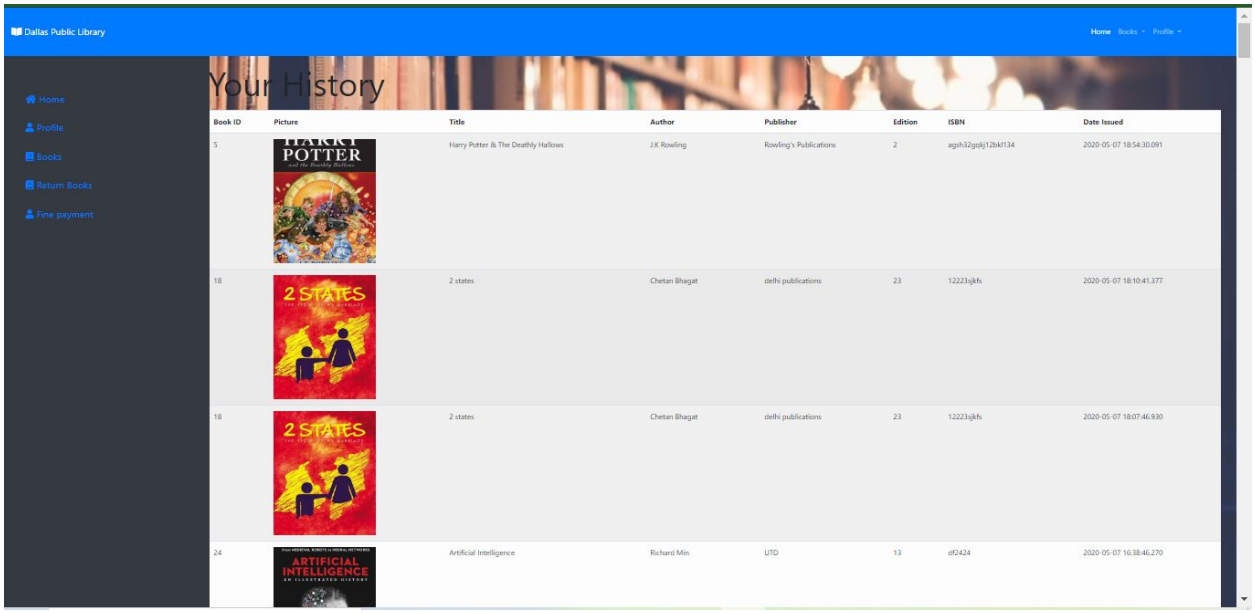
11. Return Books

In order to return books, the user needs to click on the return books button.



12. HISTORY

All the books that the particular user has borrowed till date can be viewed in ‘Your History’.



13. Exception Handling

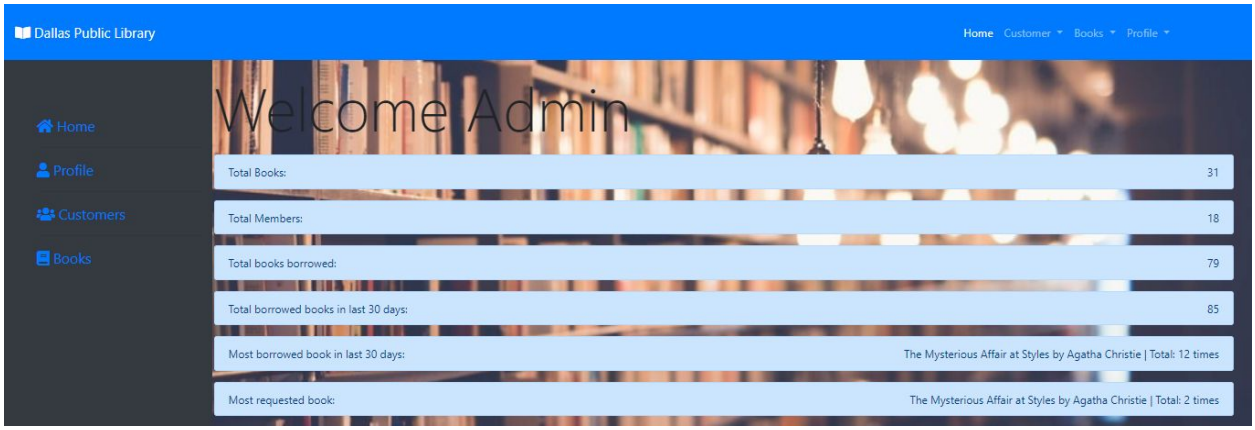
Books - A customer must select the books to be added to cart, otherwise an error will be displayed.

Cart - A customer must select a book to be deleted from cart or borrowed from cart. In case the customer doesn’t select any book and clicks on Checkout or confirm remove books the same page will be redirected and no exception would pop up.

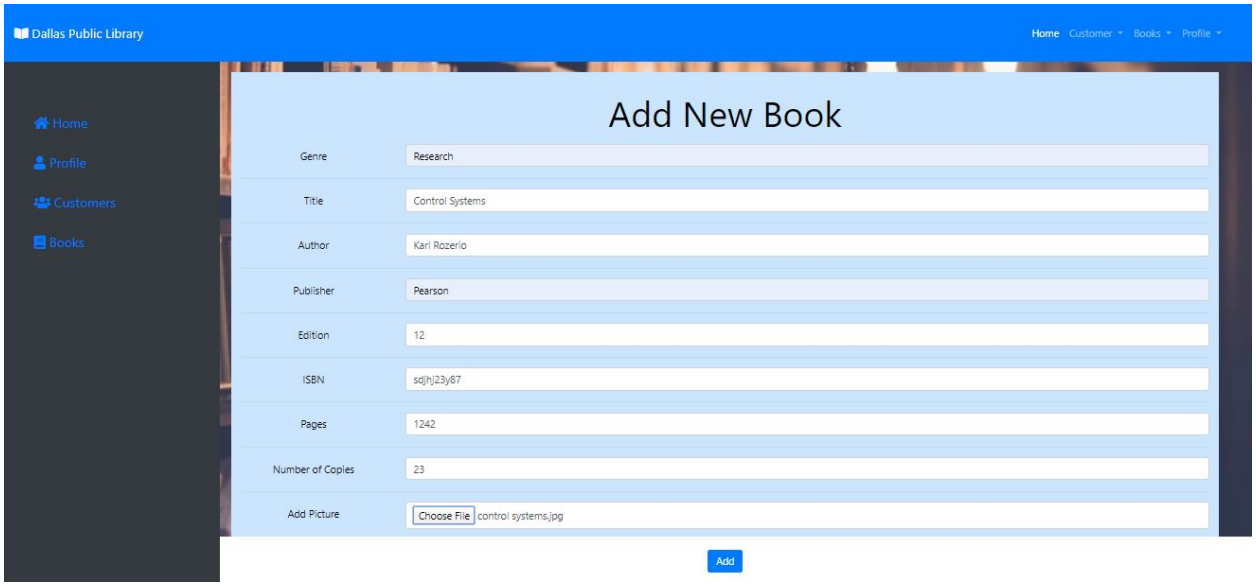
Return Books - When the return books page is displayed and in case the customer does not select a book to be returned no exception occurs and is handled in Node.js.

ADMIN

1. Welcome Page for Admin

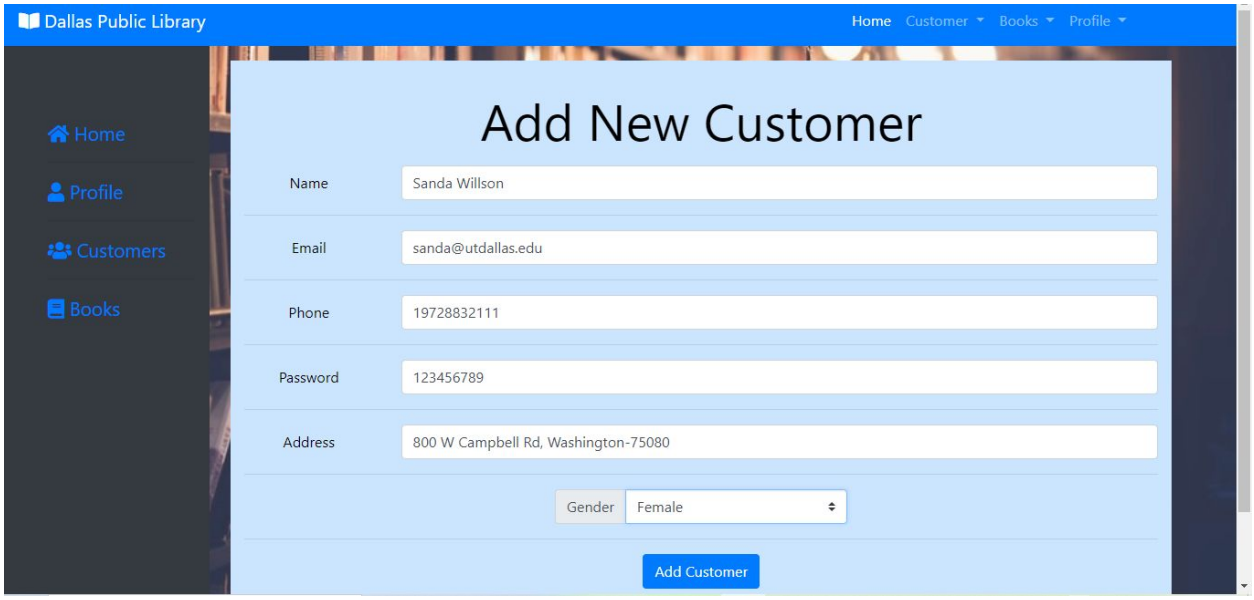


2. **Adding new book -** Admin can add new book to the database by entering all the details of the books, and then book is accessible by the customer.

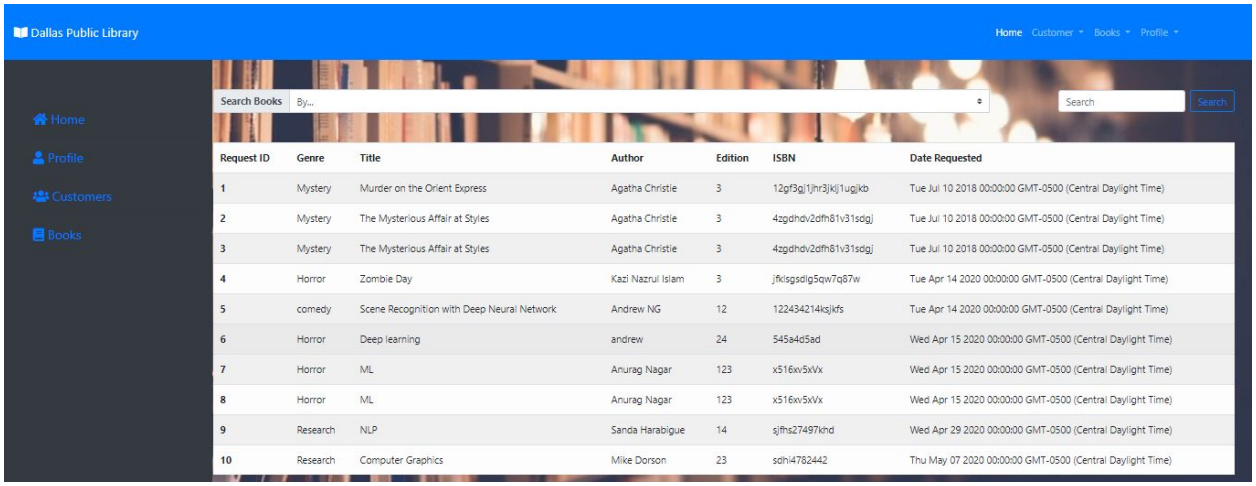


3. **Add New Customer**

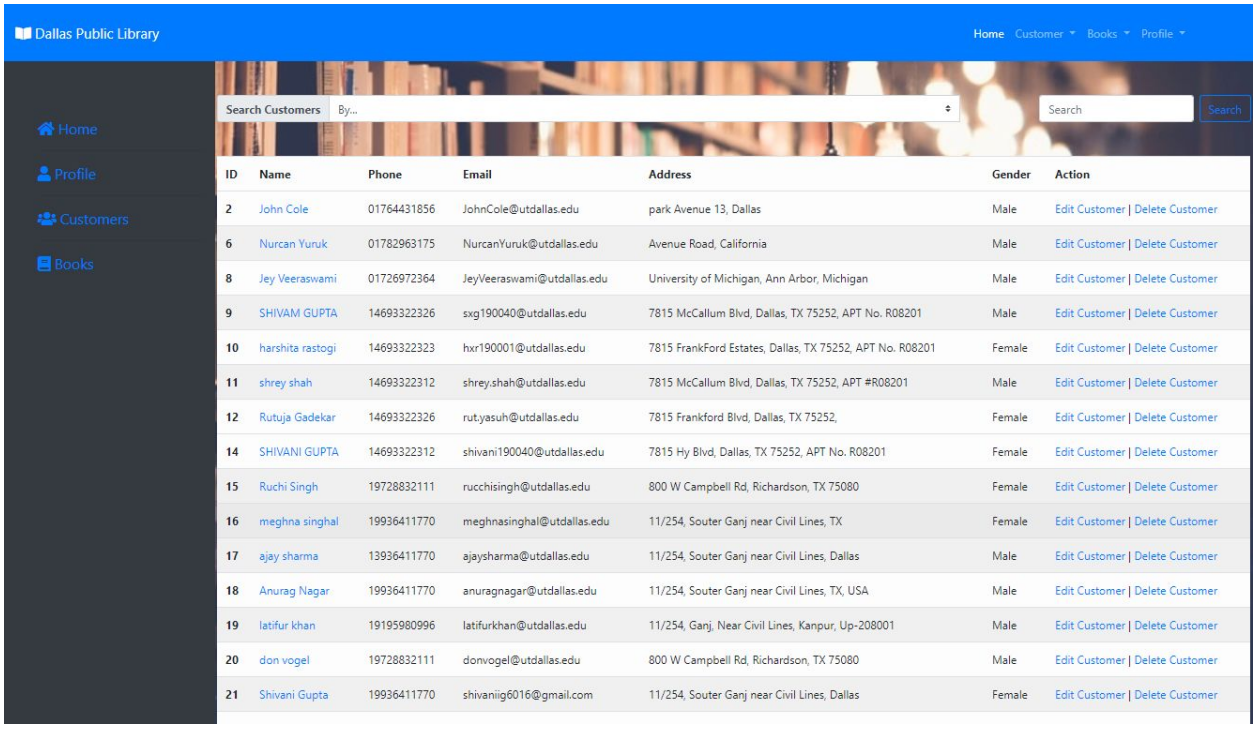
Admin can add new customer to the database, by adding new Customer functionality**.**



4. **Viewing all books by Admin**
   Admin can view all the books that stored in the database.

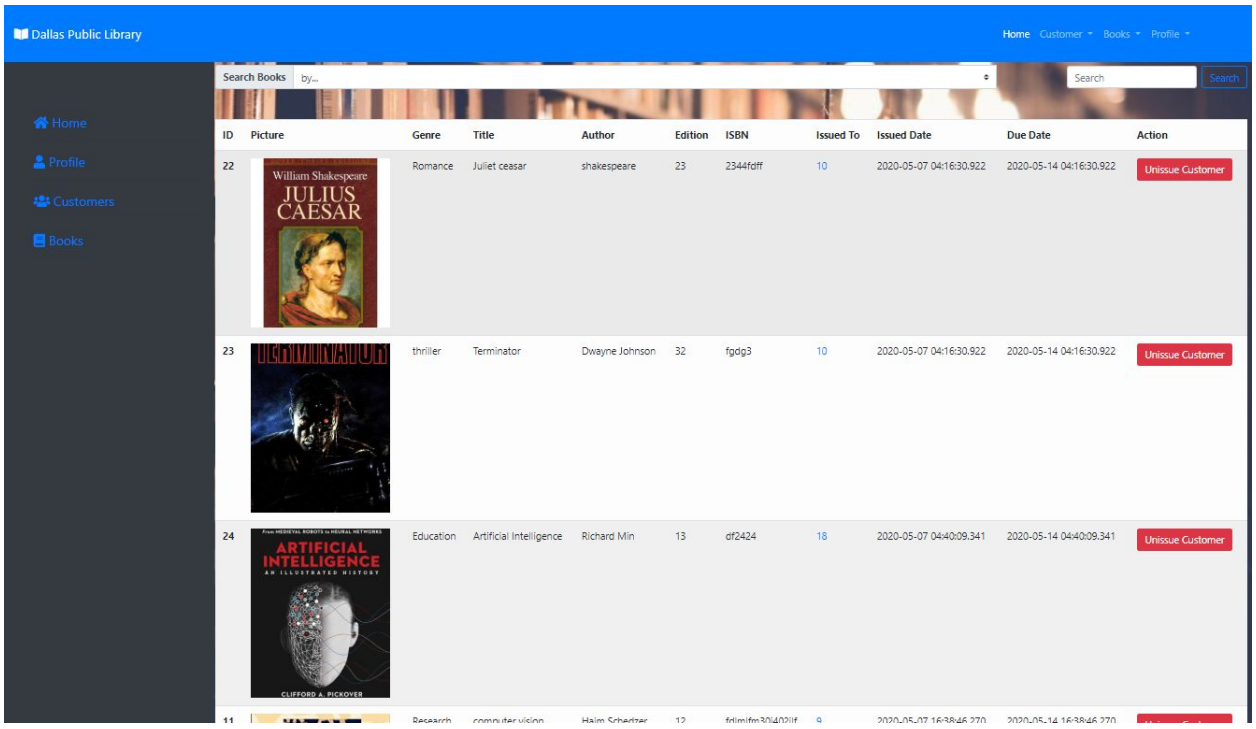


5. **Viewing all Customers by Admin**



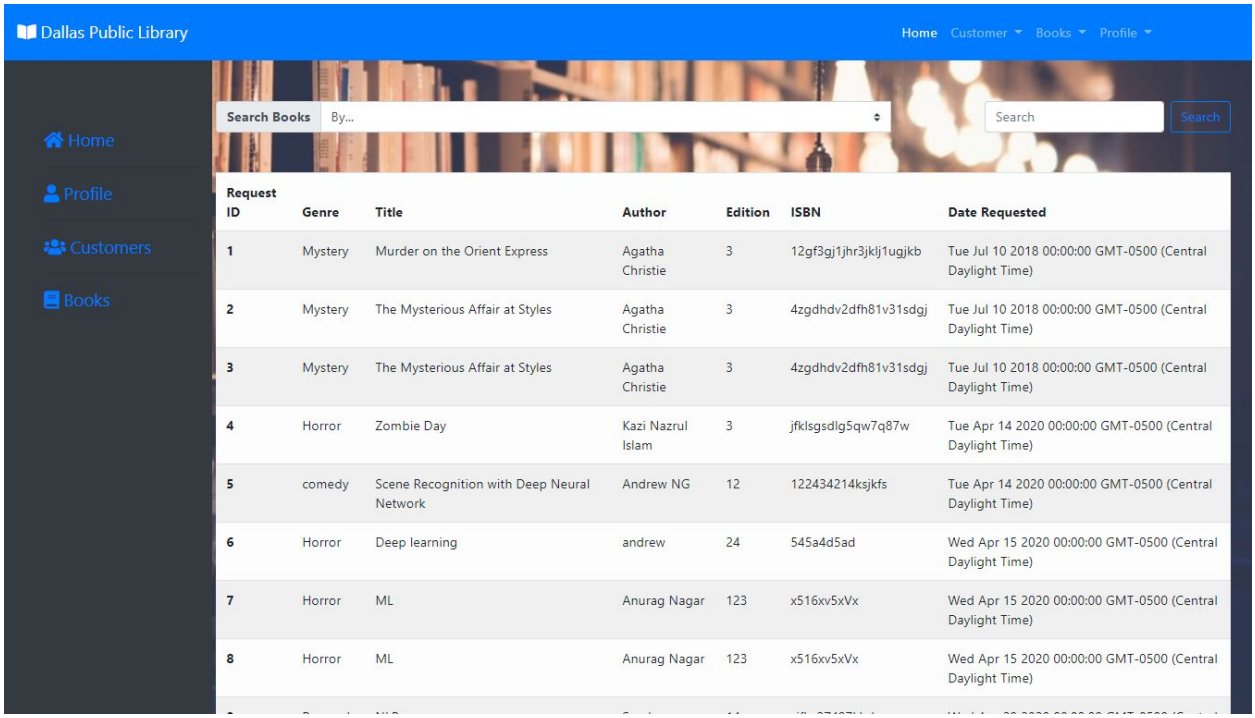
ID	Name	Phone	Email	Address	Gender	Action
2	John Cole	01764431856	JohnCole@utdallas.edu	park Avenue 13, Dallas	Male	Edit Customer | Delete Customer
6	Nurcan Yuruk	01782963175	NurcanYuruk@utdallas.edu	Avenue Road, California	Male	Edit Customer | Delete Customer
8	Jey Veeraswami	01726972364	JeyVeeraswami@utdallas.edu	University of Michigan, Ann Arbor, Michigan	Male	Edit Customer | Delete Customer
9	SHIVAM GUPTA	14693322326	sxg190040@utdallas.edu	7815 McCallum Blvd, Dallas, TX 75252, APT No. R08201	Male	Edit Customer | Delete Customer
10	harshita rastogi	14693322323	hxr190001@utdallas.edu	7815 FrankFord Estates, Dallas, TX 75252, APT No. R08201	Female	Edit Customer | Delete Customer
11	shrey shah	14693322312	shrey.shah@utdallas.edu	7815 McCallum Blvd, Dallas, TX 75252, APT #R08201	Male	Edit Customer | Delete Customer
12	Rutuja Gadekar	14693322326	rut.yasuh@utdallas.edu	7815 Frankford Blvd, Dallas, TX 75252,	Female	Edit Customer | Delete Customer
14	SHIVANI GUPTA	14693322312	shivani190040@utdallas.edu	7815 Hy Blvd, Dallas, TX 75252, APT No. R08201	Female	Edit Customer | Delete Customer
15	Ruchi Singh	19728832111	rucchisingh@utdallas.edu	800 W Campbell Rd, Richardson, TX 75080	Female	Edit Customer | Delete Customer
16	meghna singhal	19936411770	meghnasinghal@utdallas.edu	11/254, Souter Ganj near Civil Lines, TX	Female	Edit Customer | Delete Customer
17	ajay sharma	13936411770	ajaysharma@utdallas.edu	11/254, Souter Ganj near Civil Lines, Dallas	Male	Edit Customer | Delete Customer
18	Anurag Nagar	19936411770	anuragnagar@utdallas.edu	11/254, Souter Ganj near Civil Lines, TX, USA	Male	Edit Customer | Delete Customer
19	latifur khan	19195980996	latifurkhan@utdallas.edu	11/254, Ganj, Near Civil Lines, Kanpur, Up-208001	Male	Edit Customer | Delete Customer
20	don vogel	19728832111	donvogel@utdallas.edu	800 W Campbell Rd, Richardson, TX 75080	Male	Edit Customer | Delete Customer
21	Shivani Gupta	19936411770	shivaniig6016@gmail.com	11/254, Souter Ganj near Civil Lines, Dallas	Female	Edit Customer | Delete Customer

6. Un-issuing Book Authority

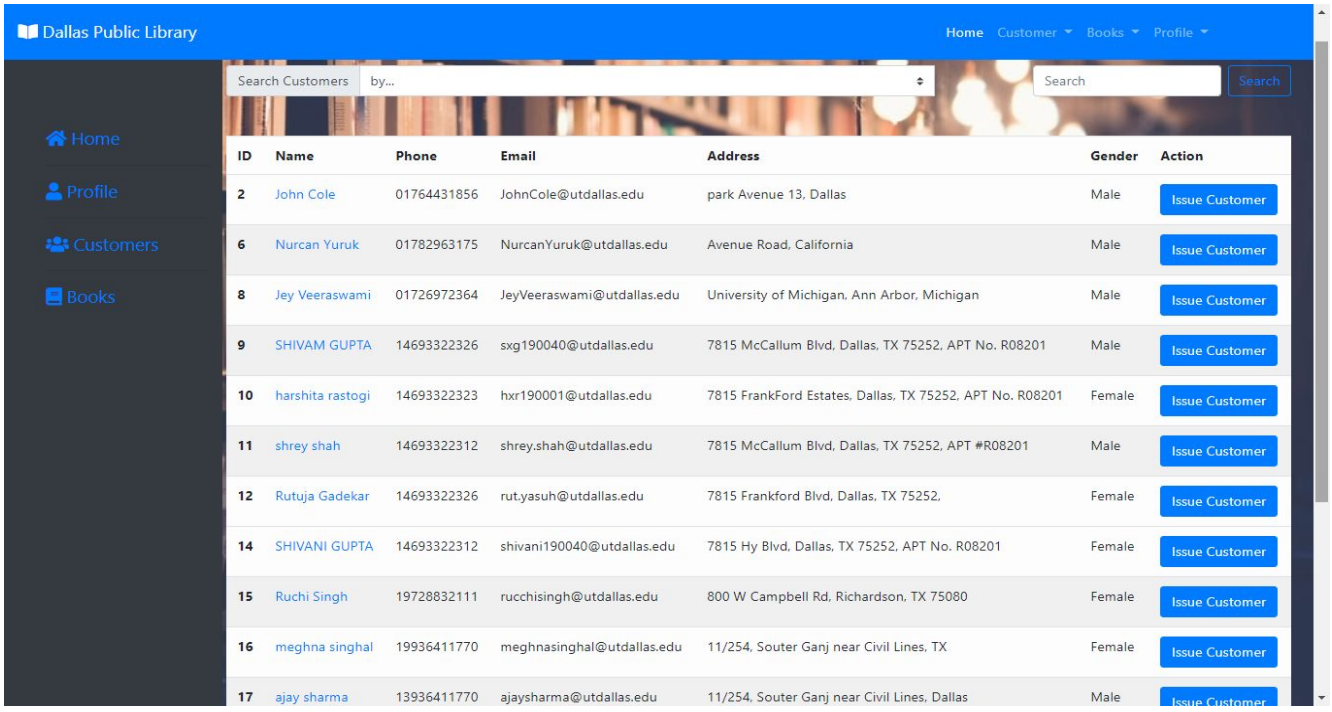


ID	Picture	Genre	Title	Author	Edition	ISBN	Issued To	Issued Date	Due Date	Action
22		Romance	Juliet ceasar	shakespeare	23	2344fdff	10	2020-05-07 04:16:30.922	2020-05-14 04:16:30.922	Unissue Customer
23		thriller	Terminator	Dwayne Johnson	32	fgdg3	10	2020-05-07 04:16:30.922	2020-05-14 04:16:30.922	Unissue Customer
24		Education	Artificial Intelligence	Richard Min	13	df2424	18	2020-05-07 04:40:09.341	2020-05-14 04:40:09.341	Unissue Customer

7. Viewing Requested Books

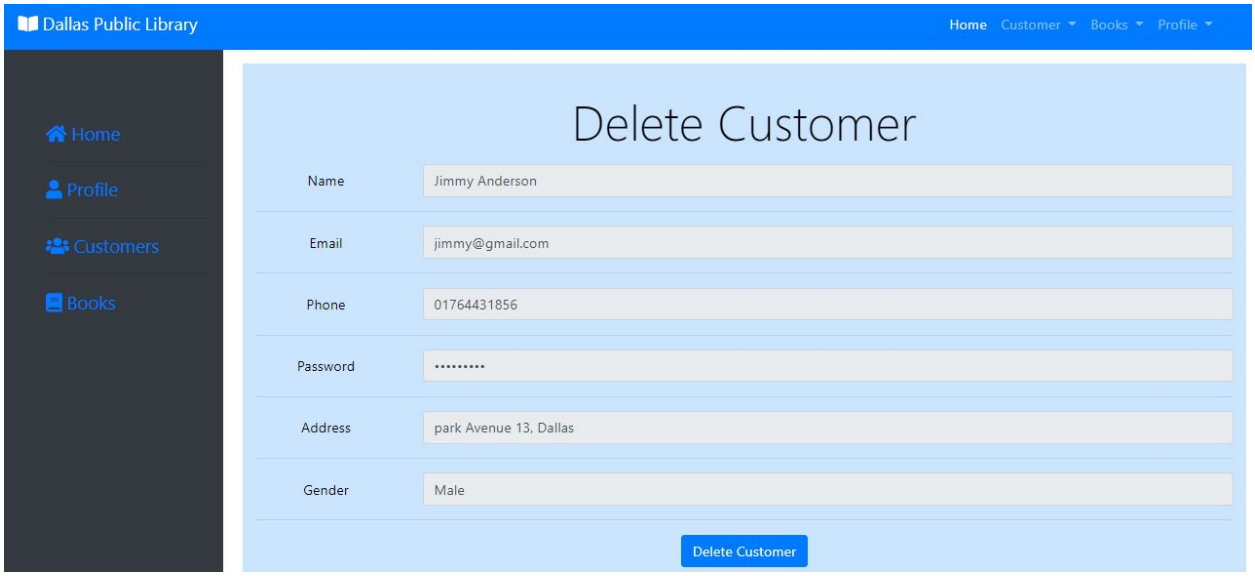


8. Issuing Book Authority to any customer



9. Soft Deleting

Case 1: Deleting the customer named Jimmy Anderson- When the customer is deleted from the Delete Customer soft delete is set to 1 in the database.

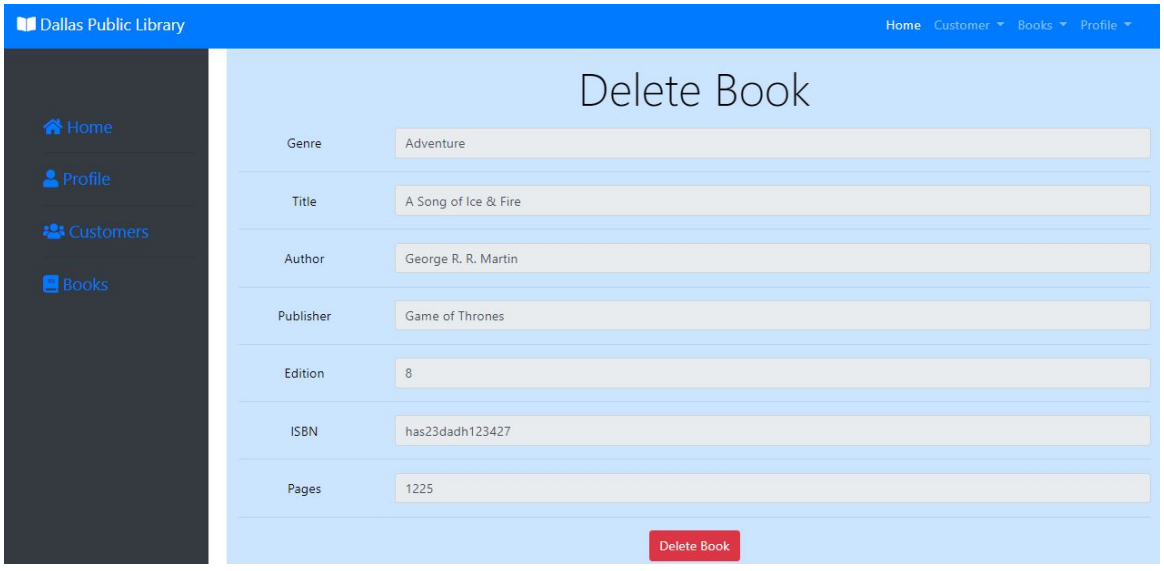


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

user_id	name	phone	email	is_admin	password	address	gender	soft_	image
2	Jimmy Anderson	01764431856	jiimy@gmail.com	0	$2b$10$DEwBuq1PHB5RrEdmOwLAmuA/764SN...	park Avenue 13, Dallas	Male	1	2_jimmy.jpeg
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Case 2:

Books can also be soft deleted. When we click delete, a book is soft deleted from the database.



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

book_id	genre	title	author	publisher	edition	isbn	pages	date_added	soft_delete	no_copies	image
3	Adventure	A Song of Ice & Fire	George R. R. Martin	Game of Thrones	8	has23dadh123427	1225	2020-05-06	1	8	GOT.jpg

Result Grid

DATABASE DESIGN

For setting up the database, we used MySQLWorkbench to create the entire schema of the library management system.

The schema is titled as **'library_management_system'**.

The connection settings are as follows:

Hostname : 127.0.0.1 Port : 3306

Username : root Password : rootroot

The relational schema is normalized to 3rd Normal Form (3NF).

NORMALIZED RELATIONAL SCHEMA

*Primary Key - **Bold**

*Foreign Key - *Italics and Underlined*

- **USERS** { **user_id**, name, phone, email, is_admin, password, address, gender, soft_delete, image }

- **BOOKS** { **book_id**, genre, title, author, publisher, edition, isbn, pages, date_added, soft_delete, no_copies, image}

- **BOOKS_REQUEST** { **request_id**, user_id, genre, title, author, edition, isbn, date }
  FOREIGN KEY (user_id) **REFERENCES** USERS (user_id)

- **CART** { **user_id, book_id**, date }
  FOREIGN KEY (user_id) **REFERENCES** USER (user_id)
  FOREIGN KEY (book_id) **REFERENCES** BOOKS (book_id)

- **ISSUE_BOOKS** { **issue_id, book_id, user_id**, issue_date, due_date, is_return }
  FOREIGN KEY (user_id) **REFERENCES** USER (user_id)
  FOREIGN KEY (book_id) **REFERENCES** BOOKS (book_id)

- **FINE** { **issue_id, book_id, user_id**, return_date, fine_amt }
  FOREIGN KEY (issue_id) REFERENCES ISSUE_BOOKS (issue_id)
  FOREIGN KEY (user_id) **REFERENCES** USER (user_id)
  FOREIGN KEY (book_id) **REFERENCES** BOOKS (book_id)

SCREENSHOT OF THE DB TABLES

USERS

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

user_id	name	phone	email	is_admin	password	address	gender	soft_	image
1	John Chang	11711568524	john@gmail.com	1	$2b$10$71szyfr7vpCABTfQEFw5suw0/37LK5n...	Marquis at waterview, Dallas	Male	0	1_John.jpeg
2	Yogesh Goyal	11764431859	yogesh@gmail.com	1	$2b$10$DEwBuq1PHB5RrEdmOwLAmuA/764SN...	UTD Campus	Male	1	2_yogesh.jpeg
3	Jennifer	11924184941	jenni@gmail.com	0	$2b$10$51IBV7Y11rtoPwIkZ/cz2.qMKVN8Sh5/X...	Northside, UTD	Female	0	3_jennifer.jpeg
4	Shreya Chopra	11723645289	shreya123@utdallas.edu	0	$2b$10$cKloHlbC0vrcIYKaJxRCWOZoyV/XxePg...	East side pradera, Dallas	Female	0	4_shreya.jpeg
5	Meghna Singhal	11711568524	mxs1800@utdallas.edu	0	$2b$10$TPuNT9i1Wj02E5.D5duTluvGw0e9TZjg...	623 Courtyards at campbell, 8102	Female	0	5_meghna.jpeg
6	Prachi Vats	14696000102	pxv180021@utdallas.edu	0	$2b$10$lhFnzcvMzMJMNiUCmWGmrenHmCk6ea...	7575 Frankford Road, 423	Female	0	6_prachi.jpeg
7	Shivam Gupta	12711568524	sxg190040@utdallas.edu	1	$2b$10$cW2Qts1ryeqPYAQfwTwZ/OIR0GK9Q...	7124 Chattham Road, 876,Dallas	Male	0	7_shivam.jpeg
8	Harshita Gupta	14696000102	hxr190001@utdallas.edu	0	$2b$10$/QSL.isYzkFGCBp2gKNKgesYrFKcKJIf/D...	7575 Frankford Road, 522	Female	0	8_harshita.jpeg

BOOKS

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

book_id	genre	title	author	publisher	edition	isbn	pages	date_added	soft_delete	no_copies	image
17	comedy	The Lord of the Rings	Andrew NG	cuba	12	jfkdsgsdw7q87w	1212	2020-05-01	0	11	The Lord of the Rings.jpg
18	comedy	2 states	Chetan Bhagat	delhi publications	23	12223sjkfs	122	2020-05-01	0	10	2states.jpeg
19	SciFi	Scene with Deep Neural Network	Anurag Nagar	IEEE	12	jfkdsgs7q87w	1221	2020-05-01	0	6	Scene with Deep Neural Network.jpg
20	comedy	Mr. Bean	Bean	Standford	24	jfkdsgsdlw7q87w	153	2020-05-01	0	3	Mr. Bean.jpg
21	Horror	scary movies	Ramanujan	UTD	32	4224ksjkfs	244	2020-05-01	0	2	scary movies.jpg
22	Romance	Juliet ceasar	shakespeare	MIT press	23	2344fdff	133	2020-05-01	0	3	Juliet ceasar.jpg
23	thriller	Terminator	Dwayne Johnson	Marvels	32	fgdg3	1331	2020-05-01	0	1	terminator.jpg
24	Education	Artificial Intelligence	Richard Min	UTD	13	df2424	134	2020-05-02	0	12	AI.jpg
25	SciFi	Avengers	Captain America	Marvels	12	dvsvs42	134	2020-05-02	0	2	avengers.jpg
26	SciFi	Ironman	Robert Downy	Marvels	23	fjsk872	1242	2020-05-02	0	4	ironman.jpg
27	SciFi	Hulk	Chris Hemsworth	Marvels	32	FFS42242VS	324	2020-05-02	0	5	hulk.jpg
28	SciFi	Black Widow	Scarlett Johnsan	cuba	13	sffsf4sgg	2313	2020-05-02	0	11	blackwidow.jpg
29	comedy	Five Point Someone	Chetan Bhagat	IEEE	23	fsfsf2442	1331	2020-05-02	0	11	5pointsomeone.jpg
30	Research	DAA	Chandra Shekhaaan	UTD	34	dfjh2424	1023	2020-05-05	0	4	DAA.jpg
31	Research	VLSI	Morris Maino	IEEE	24	124244ksjkfs	2421	2020-05-05	0	2	VLSI.jpg
32	Research	Digital Electronics	Morris Maino	Pearson	42	jfkdsgsdlw7q87w	424	2020-05-05	0	3	DE.jpg
33	Analytics	Big data Analytics	Latifur Khan	UTD	42	jfkdsgsw7q87w	1029	2020-05-05	0	3	Bigdata.jpg
34	Research	Database Design	Nurcan Yuruk	UTD	24	fdjhu283	24	2020-05-06	0	6	DB.jpg
35	Computer science	Distributed Systems	Tim Macmahan	Pearson	13	jfkdsgsd7q87w	1292	2020-05-06	0	4	Distributed Systems.jpg
36	Computer science	Distributed Systems	Tim Cook	TMH	76	jjndsd67	1202	2020-05-06	0	6	Distributed Systems.jpg
37	Educational	Data Representation	Karl Mathew	Pearson	23	sfjhj2424	1283	2020-05-07	0	4	data representation.jpg
38	Research	Control Systems	Karl Rozerio	Pearson	12	sdjhj23y87	1242	2020-05-07	0	23	control systems.jpg

books 1 | Apply | Revert

BOOKS_REQUEST

request_id	user_id	genre	title	author	edition	isbn	date
1	2	Mystery	Murder on the Orient Express	Agatha Christie	3	12gf3gj1jhr3jklj1ugjkb	2018-07-10
2	5	Mystery	The Mysterious Affair at Styles	Agatha Christie	3	4zgdhdv2dfh81v31sdgj	2018-07-10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CART

user_id	book_id	date
9	3	2020-05-07 18:06:57
9	4	2020-05-07 18:06:57
9	18	2020-05-07 19:33:07
9	19	2020-05-07 19:33:07
9	21	2020-05-07 19:33:07
9	22	2020-05-07 19:33:07
9	28	2020-05-07 19:31:00
9	29	2020-05-07 19:31:00
9	30	2020-05-07 19:31:00
9	31	2020-05-07 19:31:00

ISSUE_BOOKS

issue_id	book_id	user_id	issue_date	due_date	is_return
3	7	9	2020-05-06 21:09:06.117	2020-05-13 21:09:06.117	False
4	3	9	2020-05-07 16:30:08.367	2020-05-14 16:30:08.367	False
5	5	9	2020-05-07 16:48:30.873	2020-05-14 16:48:30.873	False
6	22	9	2020-05-07 16:48:30.873	2020-05-14 16:48:30.873	False

FINE

issue_id	book_id	user_id	return_date	fine_amt
4	3	9	NULL	0.000
5	5	9	NULL	0.000
6	22	9	NULL	0.000
NULL	NULL	NULL	NULL	NULL

CREATE TABLE STATEMENTS

USERS

```
CREATE TABLE `users` (
  `user_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(300) NOT NULL,
  `phone` varchar(11) NOT NULL,
  `email` varchar(300) NOT NULL,
  `is_admin` tinyint(1) NOT NULL,
  `password` varchar(300) NOT NULL,
  `address` varchar(300) NOT NULL,
  `gender` varchar(300) NOT NULL,
  `soft_delete` varchar(10) DEFAULT '0',
  `image` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

BOOKS

```
CREATE TABLE `books` (
  `book_id` int NOT NULL AUTO_INCREMENT,
  `genre` varchar(300) NOT NULL,
  `title` varchar(300) NOT NULL,
  `author` varchar(300) NOT NULL,
  `publisher` varchar(300) NOT NULL,
  `edition` int NOT NULL,
  `isbn` varchar(100) NOT NULL,
  `pages` int NOT NULL,
  `date_added` date DEFAULT NULL,
  `soft_delete` varchar(45) NOT NULL DEFAULT '0',
  `no_copies` int NOT NULL DEFAULT '100',
  `image` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`book_id`)
) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

BOOKS_REQUEST

```
CREATE TABLE `books_request` (
  `request_id` int NOT NULL AUTO_INCREMENT,
  `user_id` int NOT NULL,
  `genre` varchar(300) NOT NULL,
  `title` varchar(300) NOT NULL,
  `author` varchar(300) NOT NULL,
  `edition` int NOT NULL,
  `isbn` varchar(100) NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`request_id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

CART

```
CREATE TABLE `cart` (
  `user_id` int NOT NULL,
  `book_id` int NOT NULL,
```

```
  `date` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`user_id`,`book_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

ISSUE_BOOKS

```
CREATE TABLE `issue_books` (
  `issue_id` int NOT NULL AUTO_INCREMENT,
  `book_id` int NOT NULL,
  `user_id` int NOT NULL,
  `issue_date` varchar(40) DEFAULT NULL,
  `due_date` varchar(40) DEFAULT NULL,
  `is_return` varchar(10) DEFAULT 'False',
  PRIMARY KEY (`issue_id`,`user_id`,`book_id`)
) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

FINE

```
CREATE TABLE `fine` (
  `issue_id` int NOT NULL,
  `book_id` int NOT NULL,
  `user_id` int NOT NULL,
  `return_date` varchar(40) DEFAULT NULL,
  `fine_amt` decimal(10,3) DEFAULT NULL,
  PRIMARY KEY (`issue_id`,`user_id`,`book_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

WORK DIVISION

All the work has been equally divided.
The report and the CSS part has been equally split and done by all the team members.

Database Design

The basic architecture and normalization was discussed by all the members together.

Functionalities

- **User Sign up**

Client Side - Shivam
Server Side - Prachi

- **User Login**
  Client Side - Prachi
  Server Side - Harshita

- **User Profile**
  Client Side - Shivam
  Server Side - Harshita

- **Search & Filtering Functionality**
  Client Side - Shivam
  Server Side - Harshita

- **Paging Functionality**
  Client Side - Prachi
  Server Side - Shivam

- **Adding Items to Cart and Checkout**
  Client Side - Harshita
  Server Side - Prachi

- **Updating Items in the Cart**
  Client Side - Shivam
  Server Side - Prachi, Harshita

- **History of Borrowed Books**
  Client Side - Prachi
  Server Side - Harshita

- **Admin**
  Client Side - Harshita
  Server Side - Shivam

- **Soft Delete**
  Client Side - Shivam
  Server Side - Prachi