**CSCI – 760 DATABASE SYSTEMS**

**Library System**

**Submitted to:**

Dr. Huanying Gu

**Presented by:**

Priya Patel: 1093468

Yash Patel: 1124832

Nidhi Desai: 1101252

1. **Introduction:**

* Library system provides essential access to books, information, ideas and education to many readers. Library system is the public library, featuring a unique combination of different neighbourhood branches.
* Library system is easy to use portal which enables users to search and reserve books. It has more than 500 books from 100 different authors. It also provides functionality to add book, add new member, member login, and librarian login.
* We used MySQL to create database. It is highly normalized schema to support online transactions. User interface provides functionality to query database for reporting purpose.

**2. ER data model design:**  
• Entities and their attributes:

|  |  |
| --- | --- |
| Entities | Attributes |
| Book | book\_date, book\_id, book\_title, book\_isbn, author\_id, publisher\_id, book\_qty |
| Publisher | publisher\_id, publisher\_name,publisher\_add, publisher\_city |
| Author | author\_id, author\_lname, author\_frname, author\_add, author\_city |
| Branch | lib\_id, lib\_name, lib\_add, lib\_city |
| Reader | reader\_id, reader\_fname, reader\_lname, reader\_email, reader\_add, reader\_city, reader\_state, reader\_zip, reader\_phone, reader\_que, reader\_ans |
| Librarian | librarian\_id, librarian\_fname, librarian\_lname, librarian\_email, librarian\_pass |

• Domain of each attribute:

1. Book:

book\_date(<today)(datetime)

book\_id(Range >1)(int)

book\_title(public)(varchar (200))

author\_id(Range >1)(int)

book\_isbn(Any 10,13 digit number)(varchar(200))

publisher\_id(Range >1)(int)

book\_qty(Range >=1)(int)

* Primary key: book\_id
* Foreign key: author\_id, publisher\_id

1. Publisher:

publisher\_id(Range >1)(int)

publisher\_name(Any character, string)(varchar(200)

publisher\_add(North America addresses)(varchar(200))

publisher\_city(North America city, Asia addresses)(varchar(200))

* Primary key: publisher\_id

1. Author:

author\_id(Range >1)(int)

author\_lname(Any character, string) (varchar(200))

author\_frname(Any character, string)(varchar(200))

author\_add(North america addresses)(varchar(200))

author\_city(North america city)(varchar(200))

* Primary key: author\_id

1. Branch

lib\_id(Range >1)(int)

lib\_name(Any character, string)(varchar(200))

lib\_add(North america addresses)(varchar(200))

lib\_city(North america city)(varchar(200))

* Primary key: lib\_id

1. Reader

reader\_id(Range >1)(int)

reader\_fname(Any character, string)(varchar(200))

reader\_lname(Any character, string)(varchar(200))

reader\_email(any email [id-contains@/./com/in](mailto:id-contains@/./com/in))(varchar(200))

reader\_add(North america addresses)(varchar(200))

reader\_city(North america city)(varchar(200))

reader\_state(states of north America, Asia)(varchar(200))

reader\_zip(6 digit number)(int)

reader\_phone(10 digit number)(int)

reader\_que(Any character, string)(varchar(200))

reader\_ans(Any character, string)(varchar(200))

* Primary key: reader\_id

1. Librarian

librarian\_id(Range >1)(int)

librarian\_fname(Any character, string) (varchar(200))

librarian\_lname(Any character, string) (varchar(200))

librarian\_email(any email [id-contains@/./com/in](mailto:id-contains@/./com/in)) (varchar(200))

librarian\_pass(character, alpha numeric)(varchar(200))

* Primary key: librarian\_id

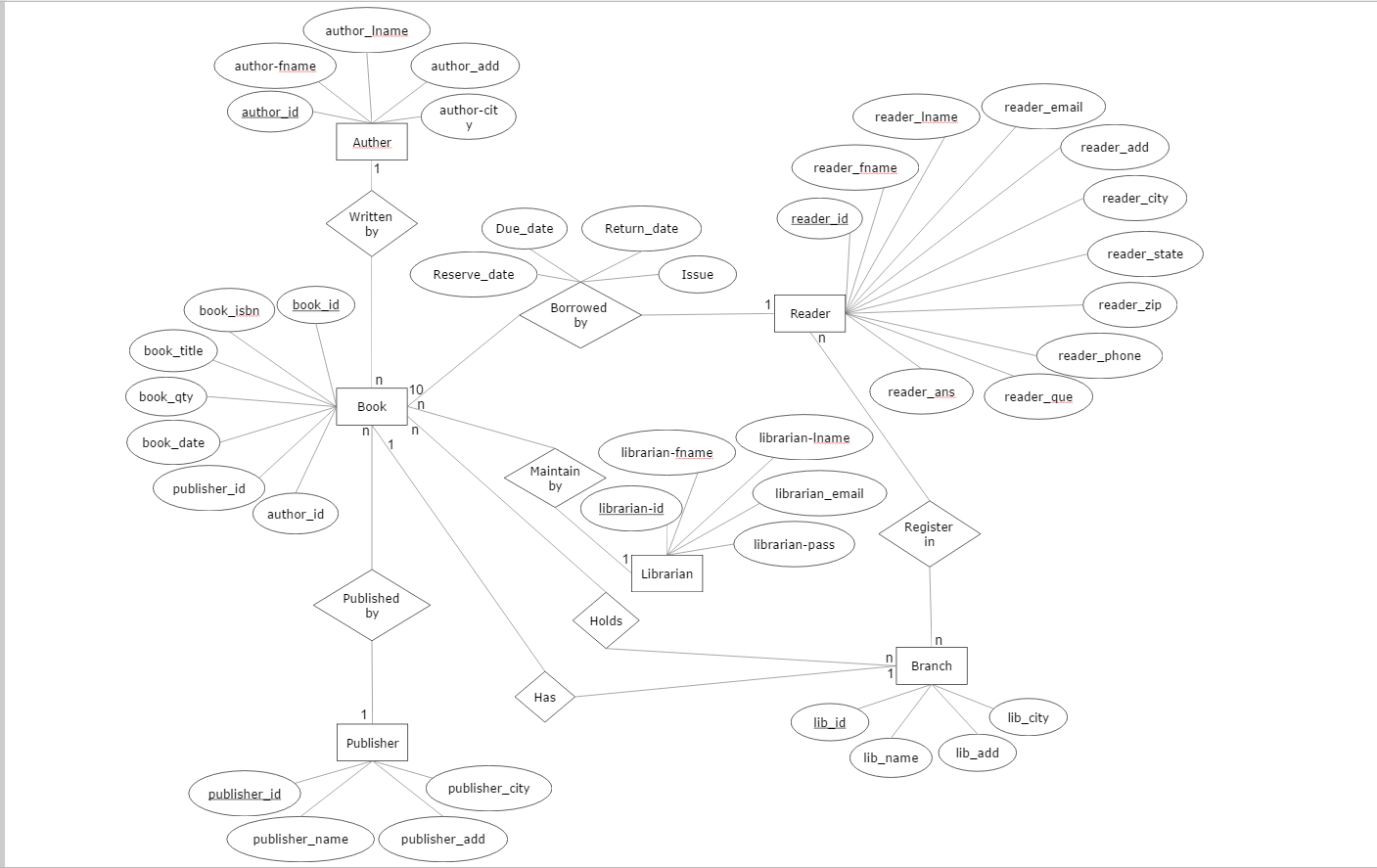
• The relationships and their attributes:

1. Librarian maintain Books
2. Book published by Publisher
3. Book written by Author
4. Branch holds Book
5. Book borrowed by Reader (Borrowed by- reserve\_date, due\_date, return\_date, Issue)
6. Reader register in Branches

* The properties of each relationship:

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship | Min-Max Constraint | Cardinality ration | Degree of the relationship |
| Librarian-Book | (1, n), (1, 1) | One to many | Binary |
| Book-Publisher | (n, 1), (1, 1) | Many to one | Binary |
| Book-Author | (n, 1), (1, 1) | Many to one | Binary |
| Branch-Book | (m, n), (n, m) | Many to many | Binary |
| Book-Reader | (10, 1), (1, 1) | Many to one | Binary |
| Reader-Branch | (m, n), (n, m) | Many to many | Binary |

* E-R diagram:



**3. Logical design of the database:**  
**1. Book:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Book\_id | Book\_title | Book\_isbn | Book\_qty | Book\_date | Publisher\_id | Author\_id | Librarian\_id |

* Primary key: book\_id
* Foreign key: librarian\_id

**2. Librarian:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Librarian\_id | Librarian\_lname | Librarian\_fname | Librarian\_email | Librarian\_pass |

* Primary key: librarian\_id

**3. Publisher:**

|  |  |  |  |
| --- | --- | --- | --- |
| Publisher\_id | Publisher\_name | Publisher\_add | Publisher\_city |

* Primary key: publisher\_id

**4. Author:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author\_id | Author\_lname | Author\_fname | Author\_add | Author\_city |

* Primary key: author\_id

**5. Branch:**

|  |  |  |  |
| --- | --- | --- | --- |
| Branch\_id | Branch\_name | Branch\_add | Branch\_city |

* Primary key: branch\_id

**6. Reader:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reader\_id | Reader\_fname | Reader\_lname | Reader\_email | Reader\_add | Reader\_city | Reader\_state | Reader\_zip | Reader\_phone | Reader\_que | Reader\_ans |

* Primary key: reader\_id

**7. Borrow:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book\_id | Reader\_id | Reserve\_date | Due\_date | Return\_date | Issue |

* Primary key: (book\_id, reader\_id)
* Foreign key 1: book\_id
* Foreign key 2: reader\_id

**8. branch\_book:**

|  |  |
| --- | --- |
| Book\_id | Librarian\_id |

* Primary key: (book\_id, librarian\_id)
* Foreign key 1: book\_id
* Foreign key 2: librarian\_id

**4. Relational Database Design:**  
After normalization:

Final relations are:

* 1. Book: book\_id, book\_isn, book\_title, book\_date
  2. Publisher: publisher\_id, publisher\_name, publisher\_add, publisher\_city
  3. Author: author\_id, author\_fname, author\_lname, author\_add, author\_city
  4. Reader: reader\_id, reader\_fname, reader\_lname, reader\_email, reader\_add, reader\_city, reader\_state, reader\_zip, reader\_ phone, reader\_que, reader\_ans
  5. Branch: lib\_id, lib\_name, lib\_add, lib\_city
  6. Librarian: librarian\_id, librarian\_fname, librarian\_lname, librarian\_email, librarian\_pass
  7. Borrow: book\_id, reader\_id, lib\_id, reserve\_id, due\_date, return\_date, issue, fine
  8. Branch\_book: book\_id, lib\_id, book\_qty, reserved\_copy
  9. Book\_pub: author\_id, book\_id, publisher\_id
  10. Lib\_branch: librarian\_id, lib\_id

**5. Implementation of Database and SQL Query:**

**1. Author table:**

//To create author table into database

CREATE TABLE `author` (

`author\_id` int(11) NOT NULL AUTO\_INCREMENT,

`author\_fname` varchar(200) NOT NULL,

`author\_lname` varchar(200) NOT NULL,

`author\_add` varchar(200) NOT NULL,

`author\_city` varchar(200) NOT NULL,

PRIMARY KEY (`author\_id`)

)

1. **Book table:** //To create book table into database

CREATE TABLE `book` (

`book\_id` int(11) NOT NULL AUTO\_INCREMENT,

`book\_title` varchar(200) NOT NULL,

`book\_isbn` varchar(200) NOT NULL,

`book\_date` date NOT NULL,

PRIMARY KEY (`book\_id`)

)

**3. Book\_pub table:**

//To create book\_pub table into database

CREATE TABLE `book\_pub` (

`book\_id` int(11) NOT NULL,

`publisher\_id` int(11) NOT NULL,

`author\_id` int(11) NOT NULL,

UNIQUE KEY `book\_id` (`book\_id`,`publisher\_id`,`author\_id`),

KEY `publisher\_id` (`publisher\_id`),

KEY `author\_id` (`author\_id`)

)

**4. borrow table:**

//To create borrow table into database

CREATE TABLE `borrow` (

`reader\_id` int(11) NOT NULL,

`book\_id` int(11) NOT NULL,

`lib\_id` int(11) NOT NULL,

`issue\_date` datetime DEFAULT NULL,

`return\_date` datetime DEFAULT NULL,

`reserve\_date` datetime DEFAULT NULL,

`due\_date` datetime DEFAULT NULL,

`fine` varchar(200) NOT NULL,

UNIQUE KEY `reader\_id` (`reader\_id`,`book\_id`,`lib\_id`),

UNIQUE KEY `reader\_id\_2` (`reader\_id`,`book\_id`,`lib\_id`),

KEY `book\_id` (`book\_id`),

KEY `lib\_id` (`lib\_id`)

)

**5. branch table:**

// To create branch table into database

CREATE TABLE `branch` (

`lib\_id` int(11) NOT NULL,

`lib\_name` varchar(200) NOT NULL,

`lib\_add` varchar(200) NOT NULL,

`lib\_city` varchar(200) NOT NULL,

PRIMARY KEY (`lib\_id`)

)

**6. branch\_book table:**

//To Create branch\_book table into database

CREATE TABLE `branch\_book` (

`lib\_id` int(11) NOT NULL,

`book\_id` int(11) NOT NULL,

`book\_qty` varchar(200) NOT NULL,

`reserved\_copy` varchar(200) NOT NULL,

UNIQUE KEY `lib\_id` (`lib\_id`,`book\_id`),

KEY `book\_id` (`book\_id`)

)

**7. librarian table:**

//To create librarian table into database

CREATE TABLE `librarian` (

`librarian\_id` int(11) NOT NULL,

`librarian\_fname` varchar(200) NOT NULL,

`librarian\_lname` varchar(200) NOT NULL,

`librarian\_email` varchar(200) NOT NULL,

`librarian\_pass` varchar(200) NOT NULL,

PRIMARY KEY (`librarian\_id`)

)

**8. lib\_branch table:**

//To create lib\_branch table into database

CREATE TABLE `lib\_branch` (

`lib\_id` int(11) NOT NULL,

`librarian\_id` int(11) NOT NULL,

KEY `lib\_id` (`lib\_id`,`librarian\_id`)

)

**9. publisher table:**

//To create publisher table into databse

CREATE TABLE `publisher` (

`publisher\_id` int(11) NOT NULL,

`publisher\_name` varchar(200) NOT NULL,

`publisher\_add` varchar(200) NOT NULL,

`publisher\_city` varchar(200) NOT NULL,

PRIMARY KEY (`publisher\_id`)

)

**10. Reader table:**

//To create reader table into database

CREATE TABLE `reader` (

`reader\_id` int(11) NOT NULL,

`reader\_fname` varchar(200) NOT NULL,

`reader\_lname` varchar(200) NOT NULL,

`reader\_email` varchar(200) NOT NULL,

`reader\_add` varchar(200) NOT NULL,

`reader\_city` varchar(200) NOT NULL,

`reader\_state` varchar(200) NOT NULL,

`reader\_zip` varchar(200) NOT NULL,

`reader\_phone` varchar(200) NOT NULL,

`reader\_que` varchar(200) NOT NULL,

`reader\_ans` varchar(200) NOT NULL,

PRIMARY KEY (`reader\_id`)

)

**Member login:**

1. User id:

For login authentication we have used below query for member login panel:

**select \* from reader WHERE reader\_id='$user\_id';**

If this query will be true than user will redirected to his member dashboard otherwise redirect to login panel again.

1. Search a book by ID, title, publisher name

select book.book\_id,book.book\_title,book.book\_isbn,book.book\_date,author.author\_id,author.author\_fname,author.author\_lname,author.author\_add,author.author\_city,publisher.publisher\_id,publisher.publisher\_name,publisher.publisher\_add,publisher.publisher\_city,branch\_book.lib\_id,branch\_book.book\_qty,branch\_book.reserved\_copy,branch.lib\_name FROM branch,book,author,publisher,book\_pub,branch\_book WHERE book.book\_id = book\_pub.book\_id AND publisher.publisher\_id=book\_pub.publisher\_id AND author.author\_id = book\_pub.author\_id AND book.book\_id = branch\_book.book\_id AND (book.book\_id='$query' OR book.book\_title like '%".$query."%' OR publisher.publisher\_name like '%".$query."%') AND branch\_book.lib\_id=branch.lib\_id

we have used above query for search book where $query argument will be posted from book search panel by member.

1. Book checkout

**In this part we have update and insert on two different tables**

insert into borrow (reader\_id,book\_id,lib\_id,issue\_date,due\_date,fine) VALUE ('$reader\_id','$book\_id','$lib\_id','$date','$due\_date',0)

UPDATE branch\_book SET book\_qty='$avail\_book',reserved\_copy='$reserved\_copy' WHERE (book\_id='$book\_id' AND lib\_id='$lib\_id')

**If the book is already reserved by that member than below query will perform on table**

UPDATE borrow SET issue\_date='$issue\_date',due\_date='$due\_date' WHERE (reader\_id='$user\_id' AND book\_id='$book\_id' AND lib\_id='$lib\_id')

1. Book return

**In this section we have perform update on table using below query**

update branch\_book set branch\_book.book\_qty='".$avail."' where (branch\_book.lib\_id='".$lib\_id."' and branch\_book.book\_id='".$\_POST['book\_id']."')

update borrow set borrow.return\_date='".date('y-m-d h:i:s')."',fine='".$\_POST['fine']."' where reader\_id='".$reader\_id."'

and book\_id='".$\_POST['book\_id']."'

1. Book reserve

In this section we have used two function **reserve and cancel**

For reseve function below query will used

UPDATE branch\_book SET reserved\_copy='$reserved\_copy' WHERE (book\_id='$book\_id' AND lib\_id='$lib\_id')

insert into borrow(reader\_id,book\_id,lib\_id,reserve\_date) values ('$reader\_id','$book\_id','$lib\_id','$date')

**For cancel function below query will used**

UPDATE branch\_book SET reserved\_copy='$reserved\_copy' WHERE (book\_id='$book\_id' AND lib\_id='$lib\_id')

DELETE FROM borrow WHERE (reader\_id='$reader\_id' AND book\_id='$book\_id' AND lib\_id='$lib\_id')

1. Compute fine for a book copy borrowed by a reader based on the current date

**For this section we have used below query**

select book.book\_title,book.book\_isbn,borrow.issue\_date,borrow.return\_date,borrow.due\_date,borrow.book\_id,borrow.lib\_id from Book, Borrow where book.book\_id=borrow.book\_id and borrow.reader\_id=".$reader\_id." and return\_date IS NULL

1. Print the list of book reserved by a reader and their status

To print and list out books which borrowed by member we used below query

SELECT book.book\_id,book.book\_title,branch.lib\_name,branch.lib\_id,borrow.reserve\_date,borrow.issue\_date,borrow.due\_date,branch\_book.reserved\_copy FROM borrow,branch,book,branch\_book WHERE borrow.reader\_id='$user\_id' AND (book.book\_id=borrow.book\_id AND branch.lib\_id=borrow.lib\_id) AND (book.book\_id=branch\_book.book\_id AND branch\_book.lib\_id=borrow.lib\_id) ORDER BY book.book\_id

**To check reserved book into cart we have used below query**

select book.book\_id,book.book\_title,book.book\_isbn,book.book\_date,author.author\_id,author.author\_fname,author.author\_lname,author.author\_add,author.author\_city,publisher.publisher\_id,publisher.publisher\_name,publisher.publisher\_add,publisher.publisher\_city,branch\_book.lib\_id,branch\_book.book\_qty,branch\_book.reserved\_copy,branch.lib\_name FROM branch,book,author,publisher,book\_pub,branch\_book WHERE book.book\_id = book\_pub.book\_id AND publisher.publisher\_id=book\_pub.publisher\_id AND author.author\_id = book\_pub.author\_id AND book.book\_id = branch\_book.book\_id AND book.book\_title like '%$query%' and branch\_book.lib\_id=branch.lib\_id

1. Print the book id and titles of books published by a publisher.

**To print book and which has to search by published name**

select book.book\_id,book.book\_title,book.book\_isbn,book.book\_date,publisher.publisher\_name,branch\_book.book\_qty,branch.lib\_name,branch.lib\_add,branch.lib\_city FROM book,book\_pub,publisher,branch\_book,branch WHERE (book.book\_id=book\_pub.book\_id AND branch.lib\_id=branch\_book.lib\_id AND branch\_book.book\_id=book.book\_id AND book\_pub.publisher\_id=publisher.publisher\_id) AND (publisher.publisher\_id = '$query' OR publisher.publisher\_name like '%".$query."%')

Librarian login:

1. Search book copy and check its status

**In this section we have used below query**

select book.book\_id,book.book\_title,book.book\_isbn,book.book\_date,author.author\_id,author.author\_fname,author.author\_lname,author.author\_add,author.author\_city,publisher.publisher\_id,publisher.publisher\_name,publisher.publisher\_add,publisher.publisher\_city,branch\_book.lib\_id,branch\_book.book\_qty,branch\_book.reserved\_copy FROM book,author,publisher,book\_pub,branch\_book WHERE book.book\_id = book\_pub.book\_id AND publisher.publisher\_id= book\_pub.publisher\_id AND author.author\_id = book\_pub.author\_id AND book.book\_id = branch\_book.book\_id AND book.book\_title like '%$query%'

1. Print branch information

**In this section we have used below query**

select \* FROM branch WHERE branch.lib\_city like '%".$query."%' OR branch.lib\_name like '%".$query."%'

1. Print top 10 most frequent borrowers in a branch and the number of books each has borrowed

select count(\*) as c,b.book\_title,br.lib\_name from borrow as bo, book as b,branch as br where bo.book\_id=b.book\_id and bo.lib\_id=br.lib\_id group by bo.book\_id order by c desc

1. Print top 10 most borrowed books in branch

select count(\*) as c,r.reader\_fname,br.lib\_name,r.reader\_email,r.reader\_phone from borrow as bo,reader as r,branch as br,book where br.lib\_id=bo.lib\_id and r.reader\_id=bo.reader\_id and book.book\_id=bo.book\_id group by bo.reader\_id order by c desc limit 0,10

1. Find the average fine paid per reader.

select avg(fine) as a,r.reader\_fname,r.reader\_lname from borrow bo,reader r where r.reader\_id=bo.reader\_id group by bo.reader\_id having avg(fine)

1. Branch search

select \* FROM branch WHERE branch.lib\_city like '%".$query."%' OR branch.lib\_name like '%".$query."%'

1. Member search

select reader.reader\_id,reader.reader\_fname,reader.reader\_lname,reader.reader\_email,reader.reader\_add,reader.reader\_city,reader.reader\_state,reader.reader\_zip,reader.reader\_phone FROM reader WHERE reader.reader\_id = '$query'

**6. Application design:**  
We have used php server side scripting language for backend use and mysql server as DBMS.

Firstly as a part of requirement gathering we have list out features and functionality as per user needs and we decided to use mysql server for maintain database and we used php because implementation is pretty easy to use and understand.

**7. User guide:**

* First go to this link: <http://yashpatel.in/lib/>
* After selecting above link, home page will appear on the screen.
* User can select one of the two available sections at one time.

1. Member login
2. Librarian login

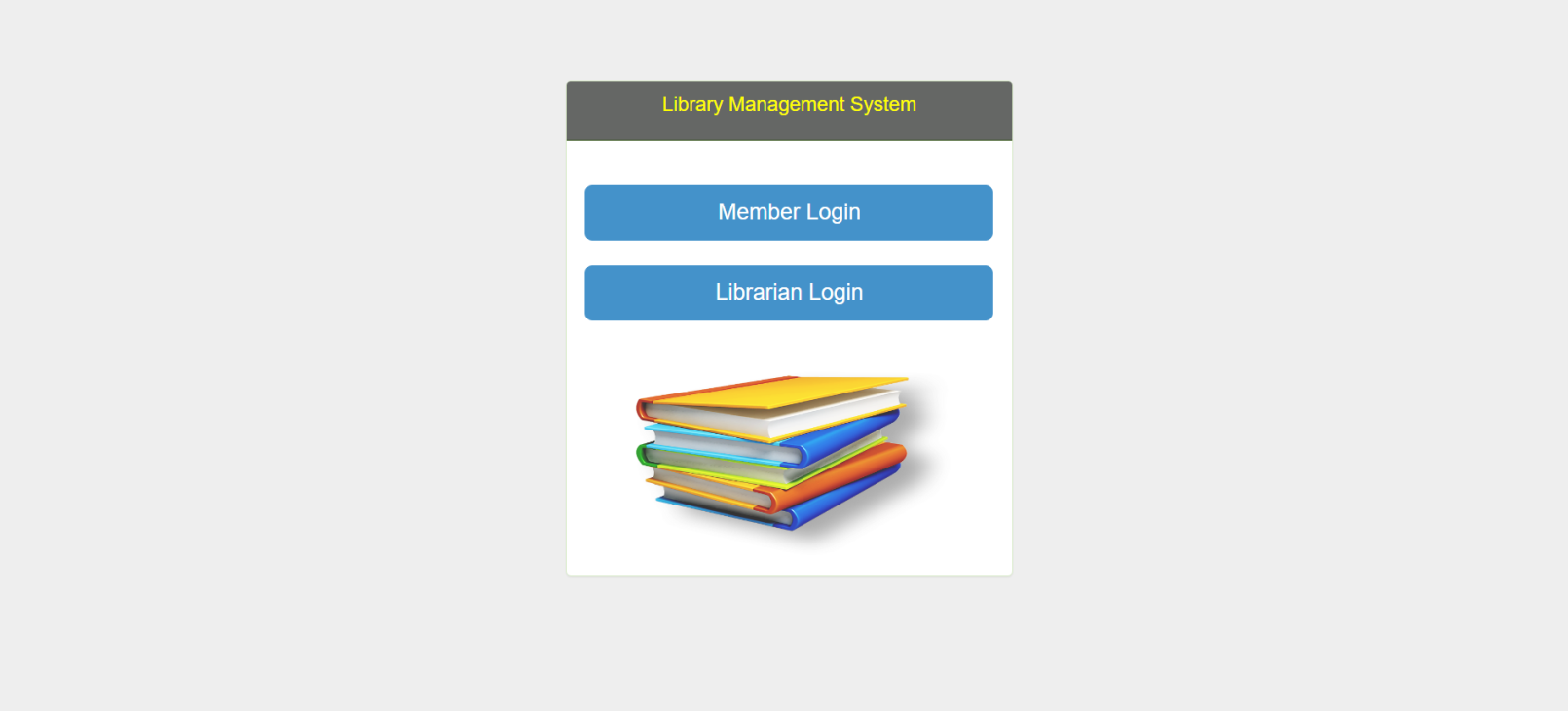
**Librarian Login:**

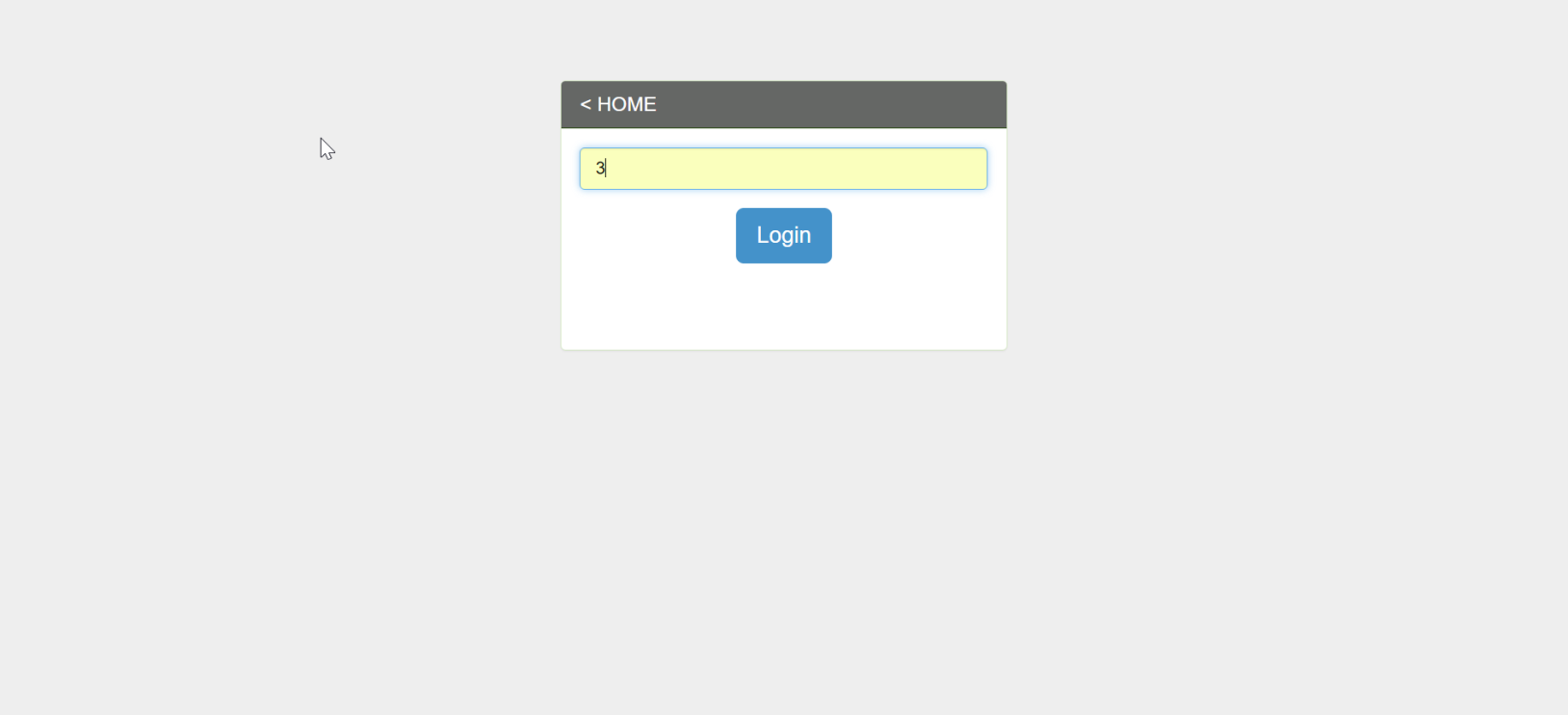
* Select librarian login. Enter email id and password. For example, you can write email id: [gunyit@gmail.com](mailto:gunyit@gmail.com) and password: Nyit123
* User can also create their own email and password:
  + <http://yashpatel.in/lib/admin/admin.php?i=1>
  + Write down all the required information to the entire field.
  + Password should be minimum 4 digits.
  + Click on register.
  + Go to librarian login and write down your generated email id and password.
* After successful librarian login, another page will appear. From that page librarian can do different things like:
  + Add any new book to the system. Enter all required information such as book title, book ISBN number, etc. Click on the add button then another form with all information you enter will appear then you can select “ok”. You can also edit the information you already entered by clicking on the edit button.
  + To go to the librarian dashboard you can click on the “<librarian dashboard”.
  + Librarian can also search available book in the library and check its status that if book is available or not. For example, write down the book name such as “algorithm concept”.
  + Librarian can add any new member to library. From this, member can get his ID. This ID can be used by member to login into their account.
  + User can also search for library information by selecting print branch information. After selecting print branch information user have to write branch location or branch name. For example, you can write Jersey City. This will give you all the information about branch like branch id, branch name, branch address and branch city.
  + Librarian can get “top 10 most frequent borrowers information” and the number of books each has borrowed.
  + Librarian can print “top 10 most borrowed books in branch.
  + Librarian can find the average fine paid per reader.
* If user want to go to home page then he/she has to be logout from his/her login. After logging out from librarian user will be directed to home page.

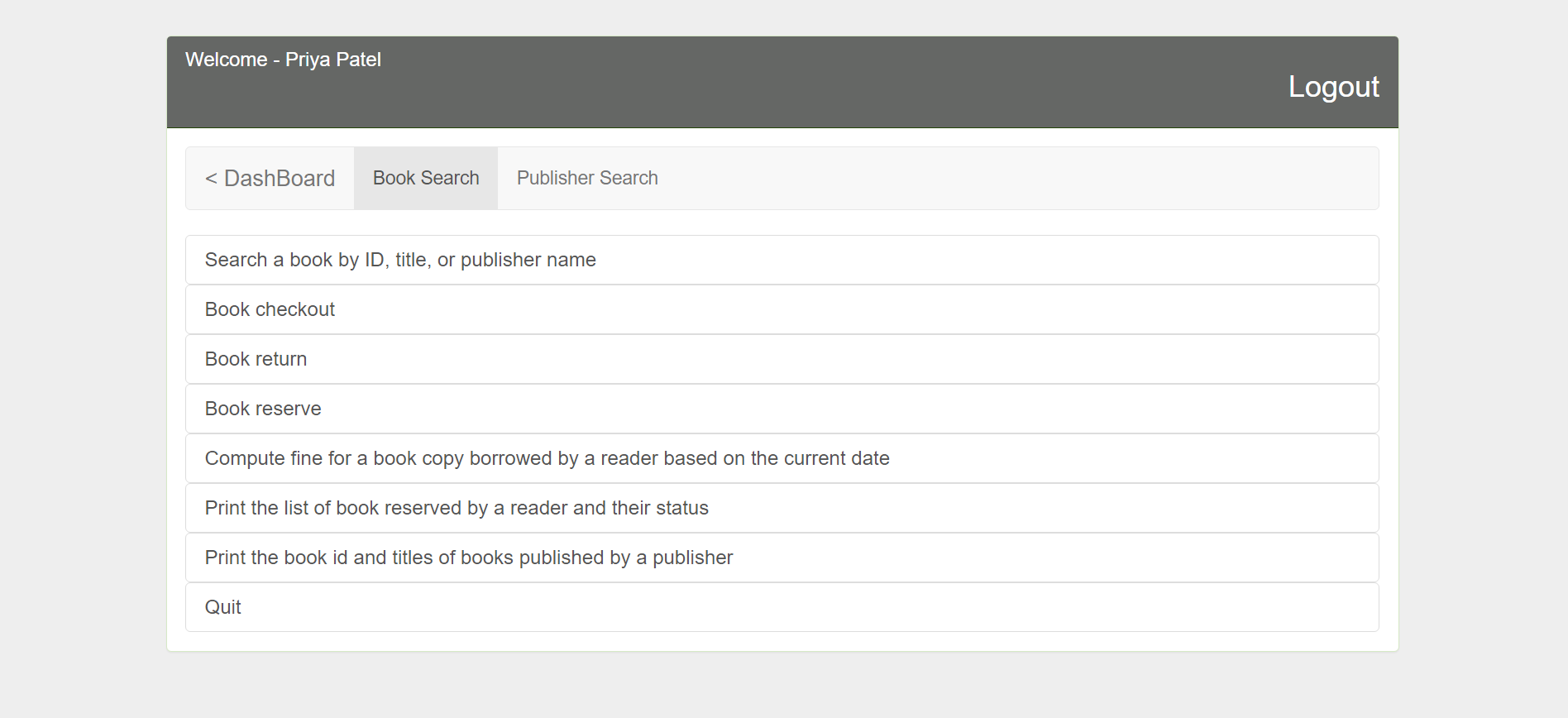
**Member Login:**

* Now, user can go to member login. Enter your member id. For example, user can write 1, 2, or 3, etc.
* After member login new page will appear so that member can do different activity such as:
  + Member can search book by book id, book title or by publisher name. For example, write book id: 1/2/3../40, write book title: fundamental of computer science, or write publisher name: the hindu.
  + After getting all information, user can go to member page to search another book by clicking on the <DashBoard section.
  + They can checkout book, return their book and also they can reserve any book.
  + System can compute fine for a book copy borrowed by a reader based on the current date.
  + Member can see the report for the list of book reserved by a reader and can also see the status for that book.
  + Member can also get the information about book id and book title by providing publisher name.
* After getting all required information member can quit that section and can come to home page.

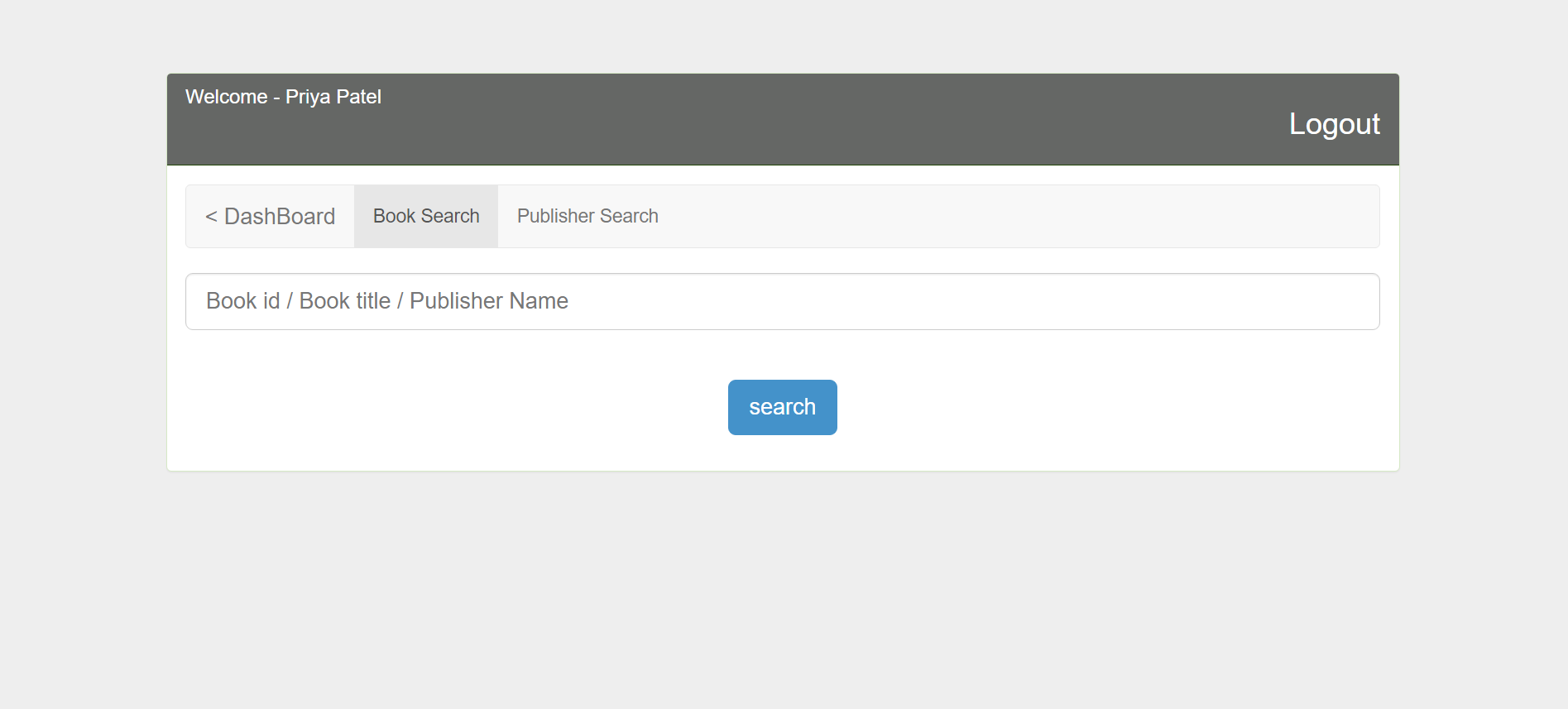
**8. Window snapshots of the use of the program for each function:  
Home page:**

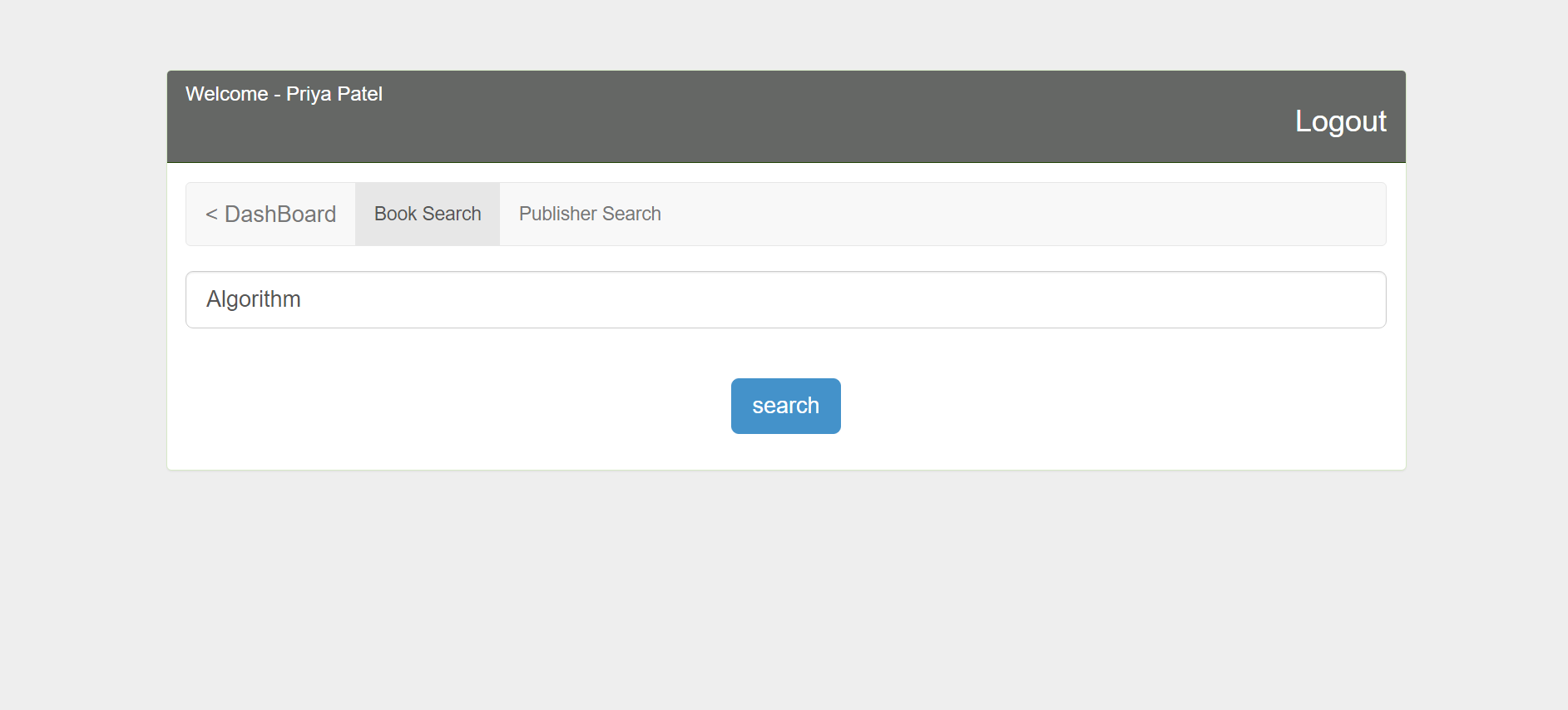
**  
Member login:**

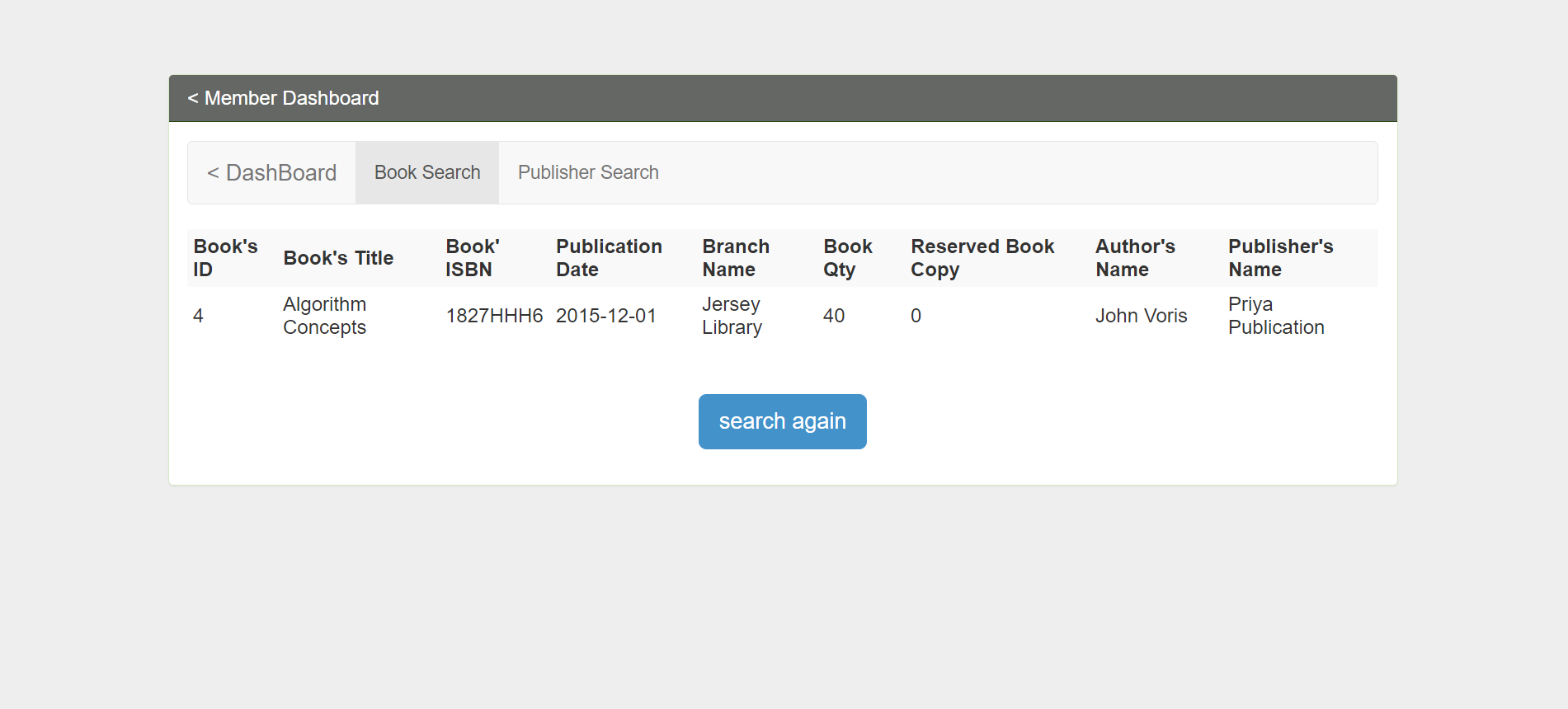
**  
Member submenu page (after successful login):**

****

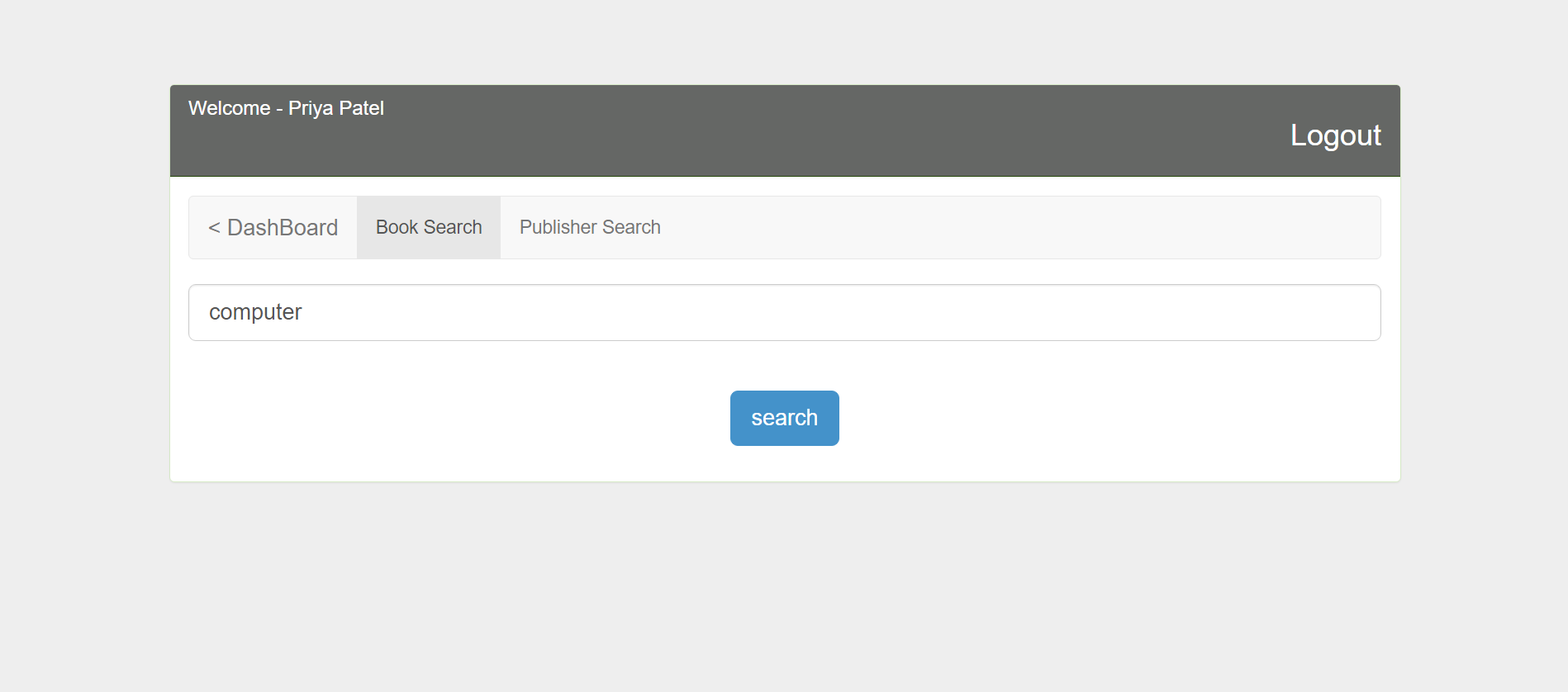
**Book search:**

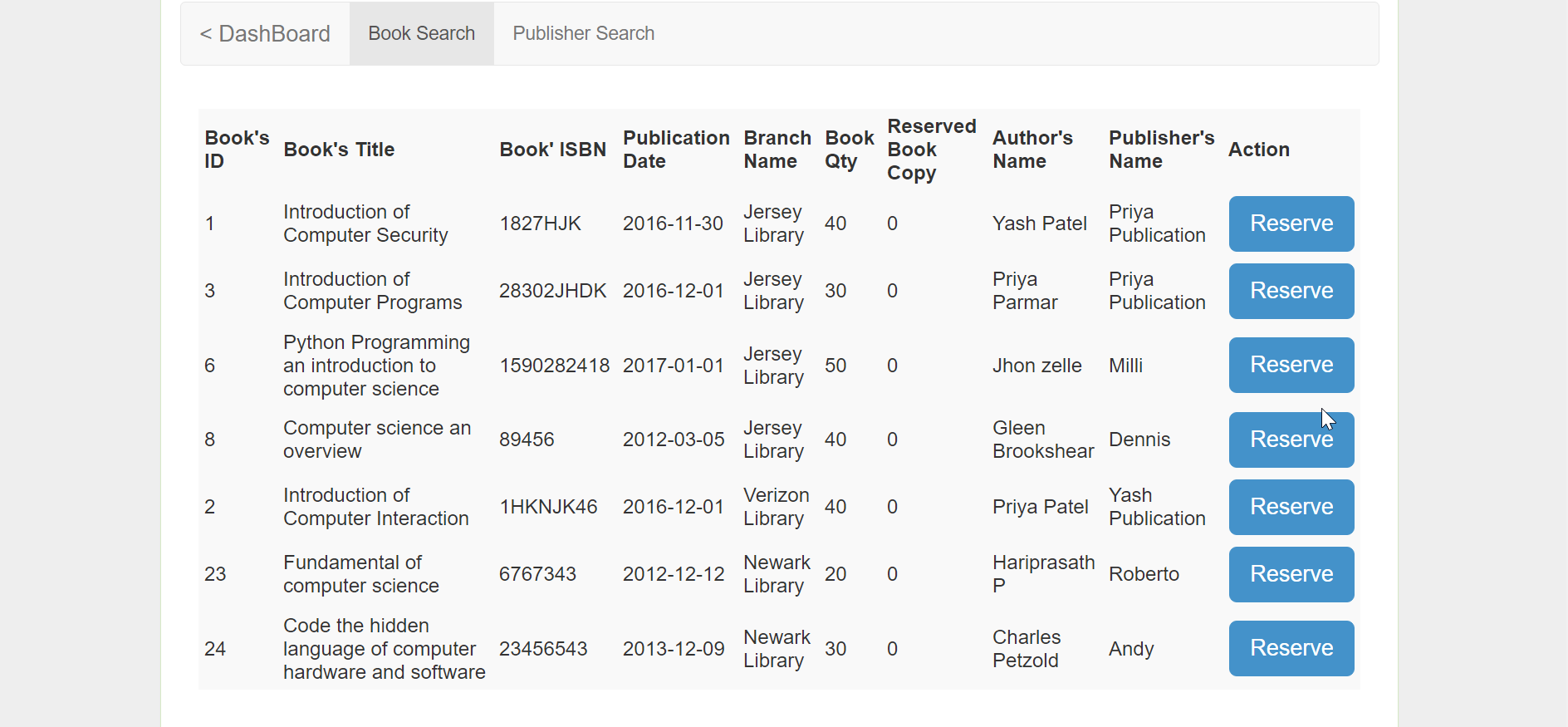
****

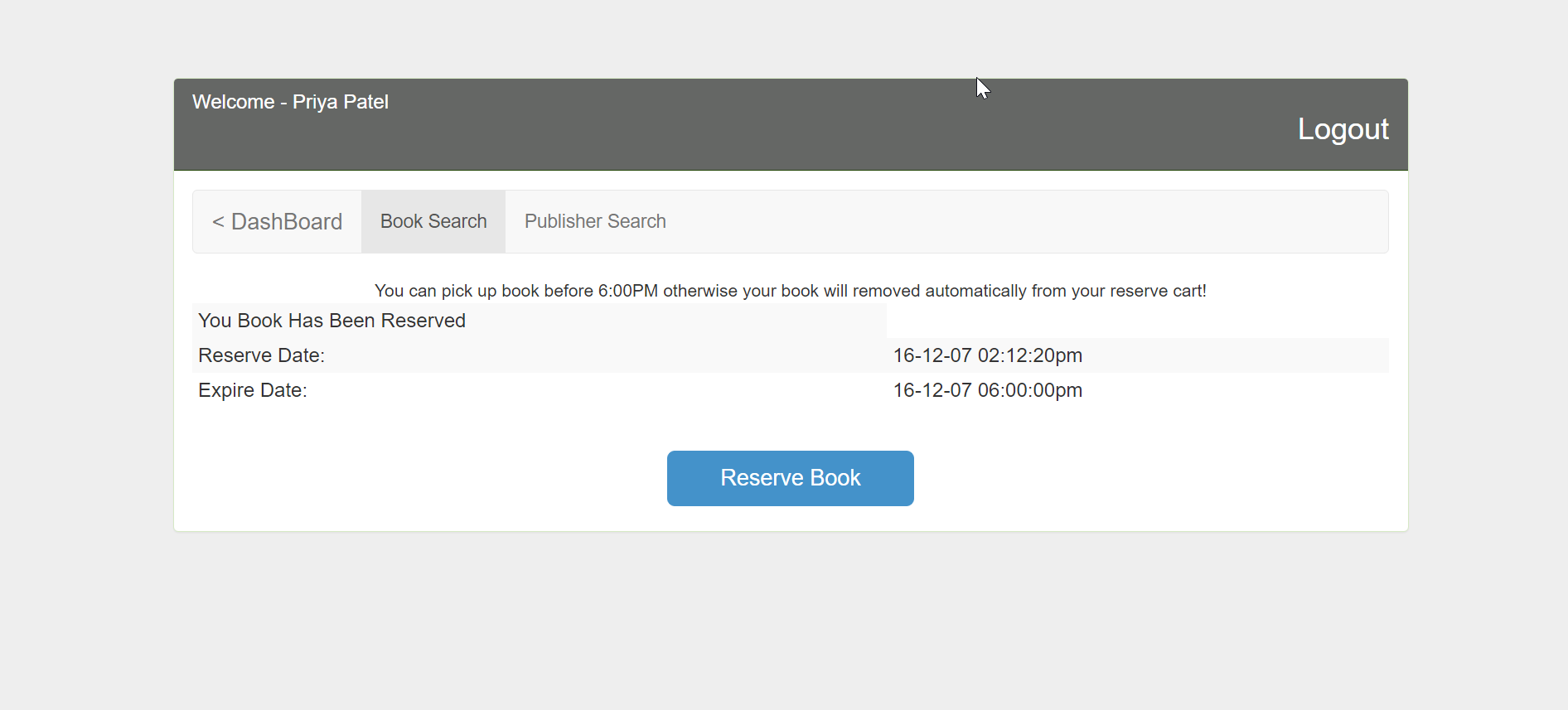
****

****

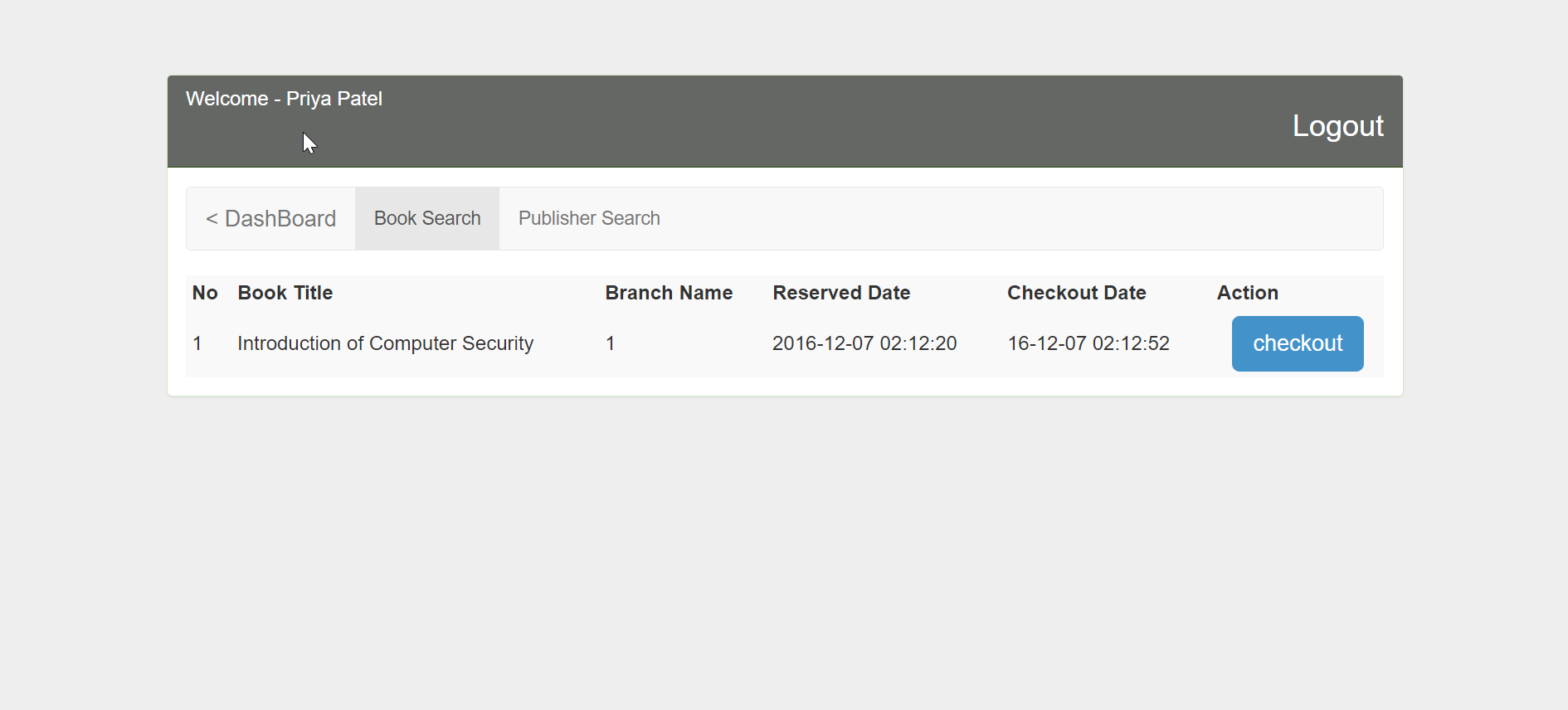
**Book Reserve:**

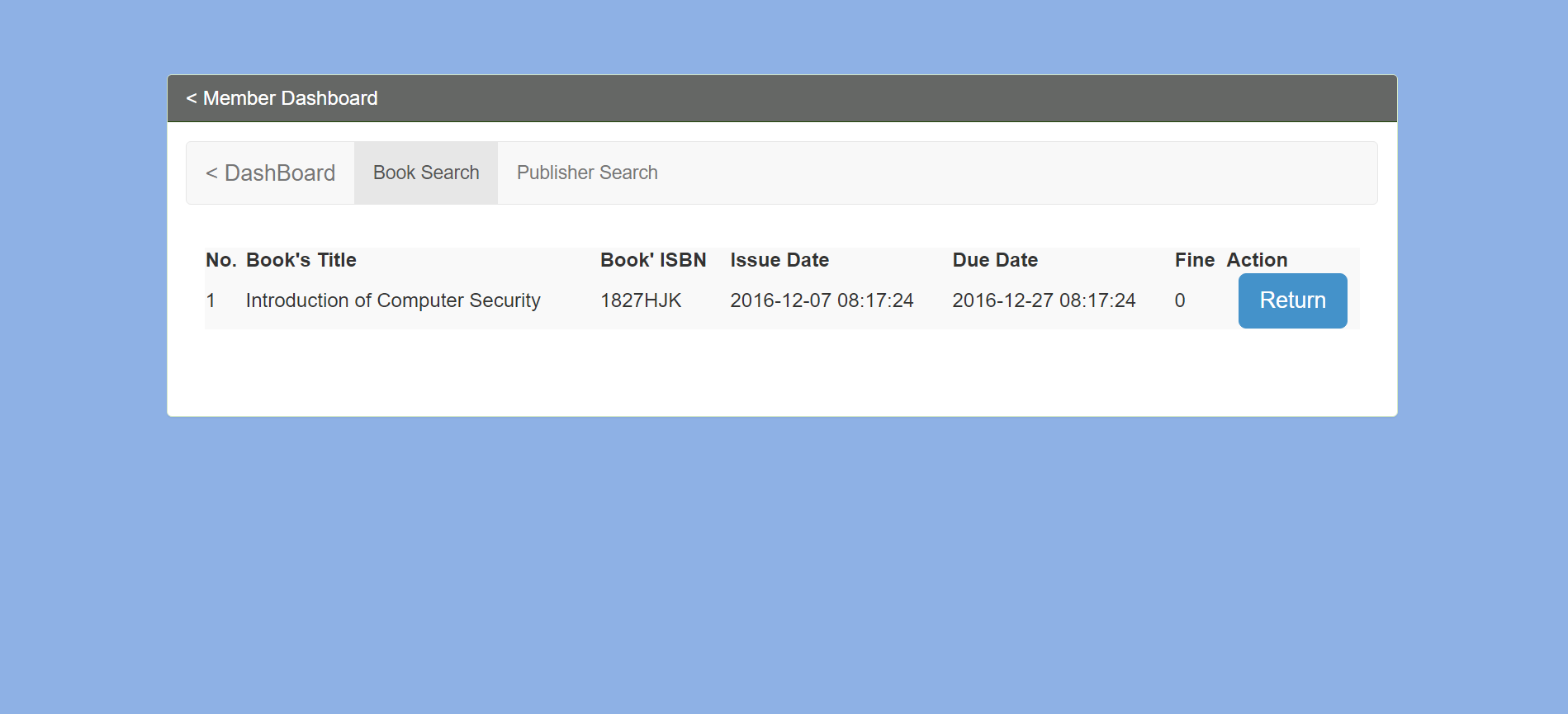
****

****

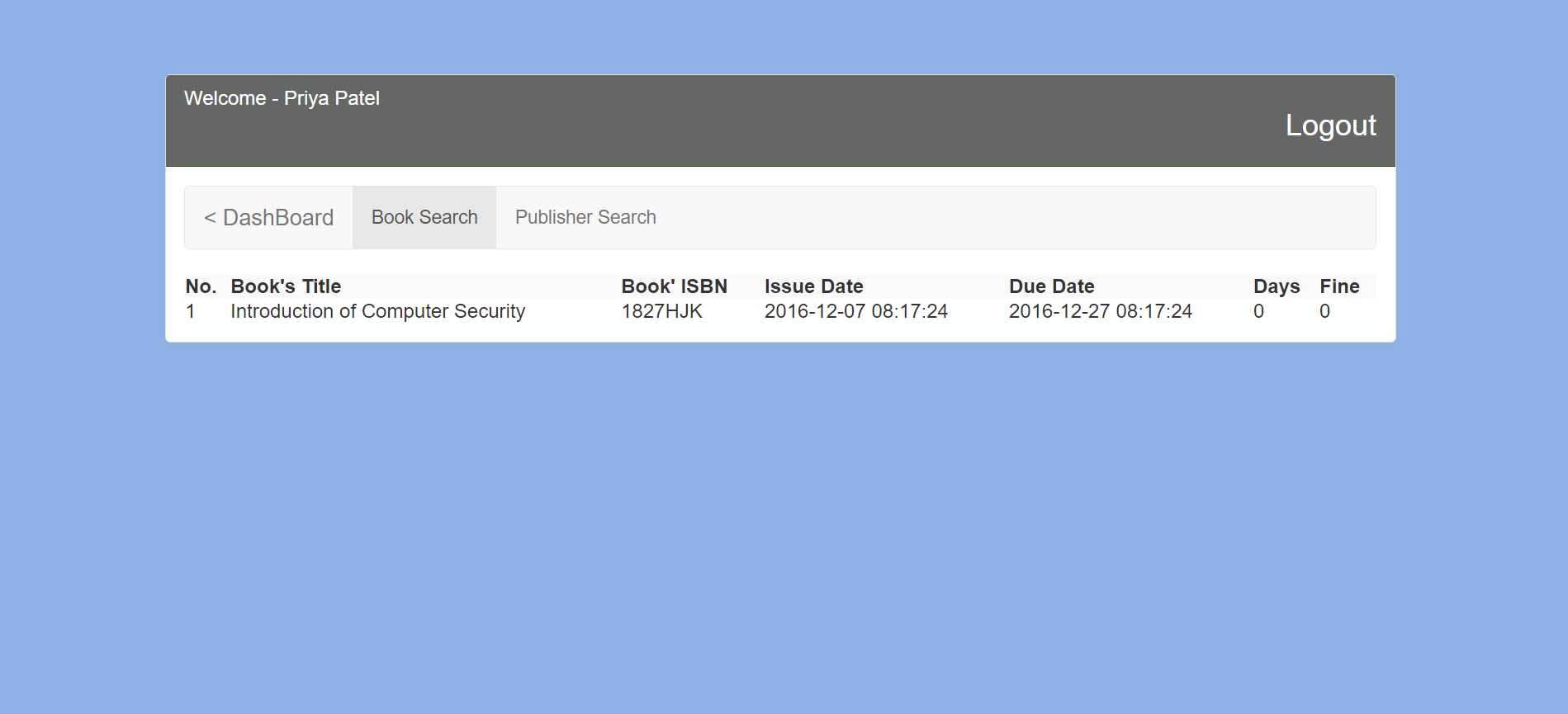
****

**Book Checkout:**

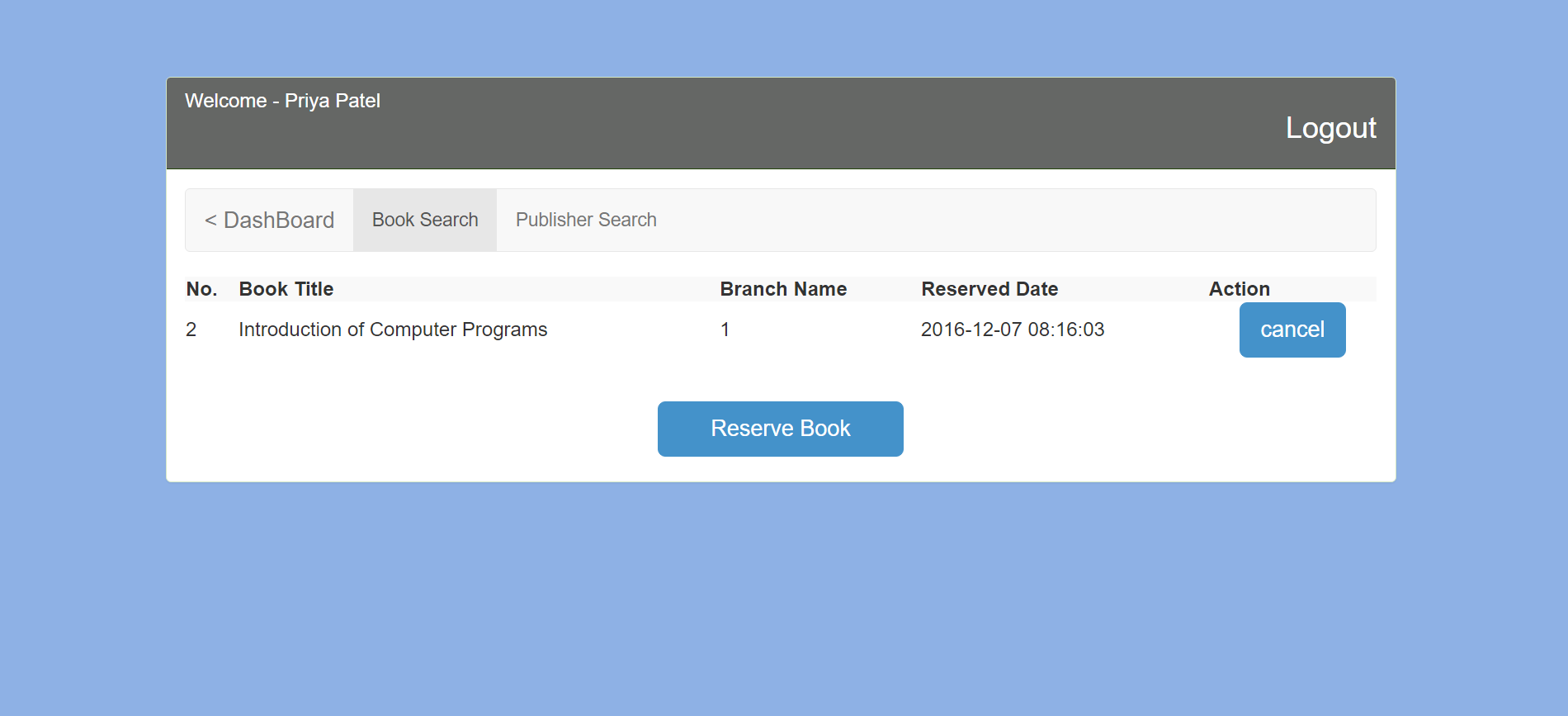
****

****

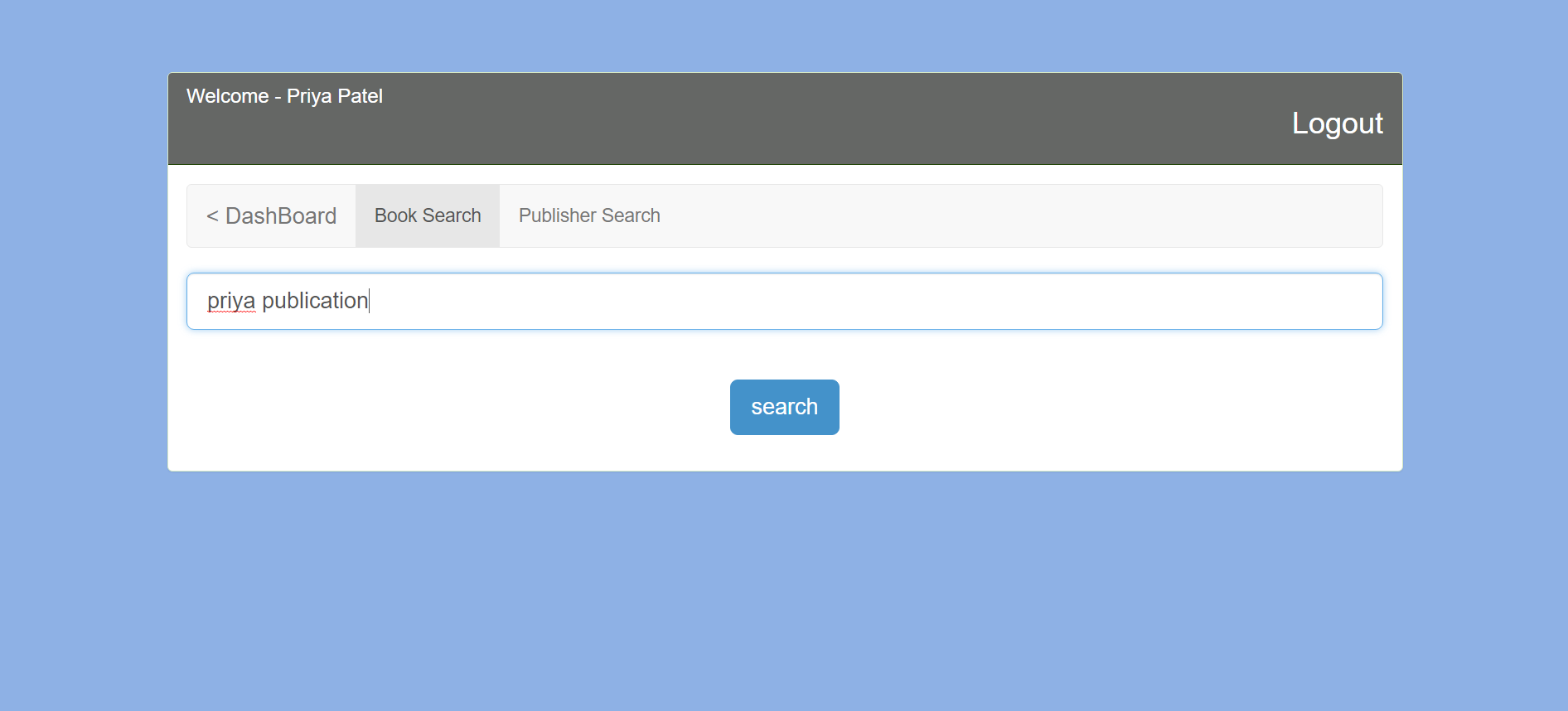
**Compute fine:**

****

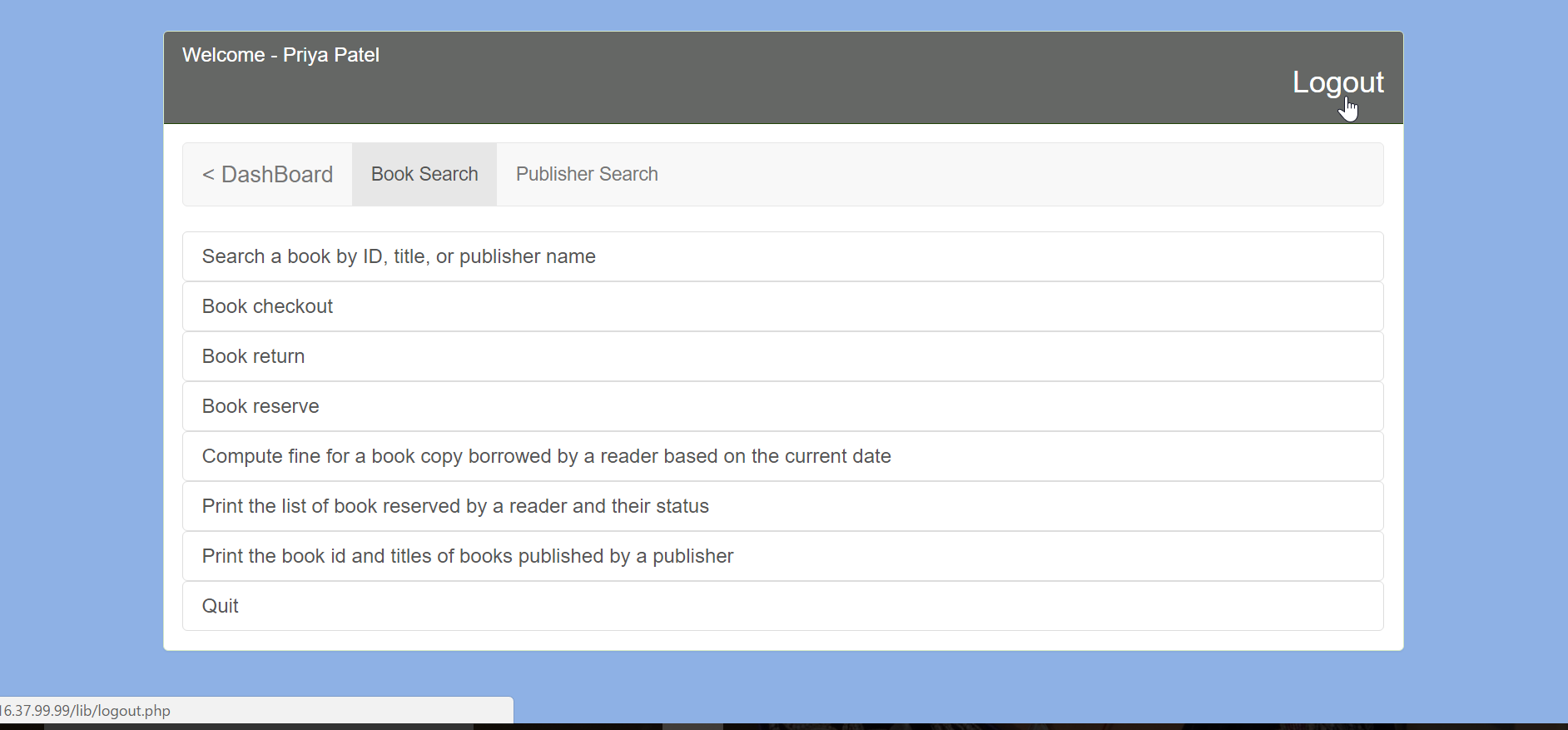
**Print the list of book copy reserved by reader:**

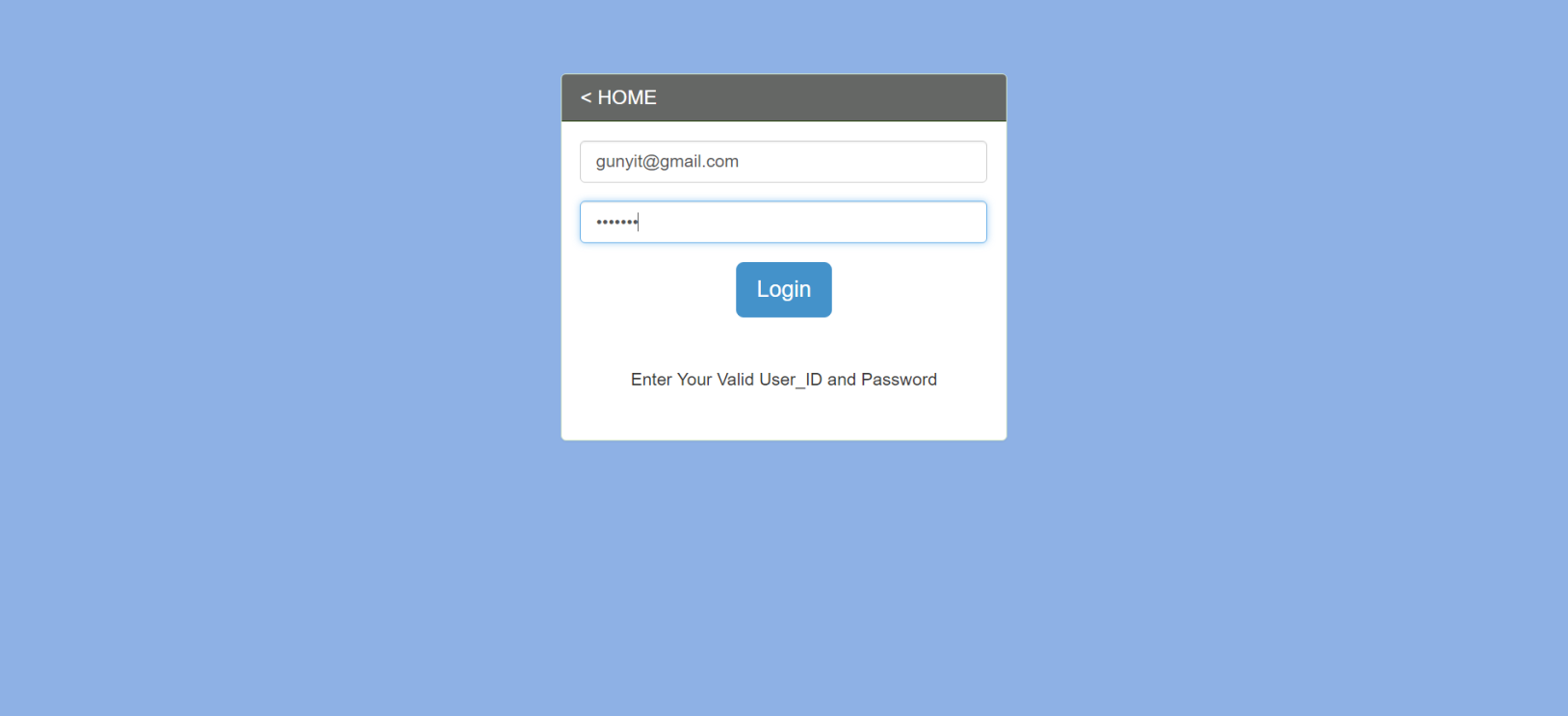
****

**Publisher search:**

****

**Logout:**

**  
Librarian login:**

****

**9. Specify in detail every team member’s work:**

Yash patel: Created source code

Priya patel: Created database and documentation

Nidhi desai: Queries and database normalization