**BookShare**

**Documentation**

****

**Table of Contents**

**Overview**…………………..…………………………………………....*p.3*

**Project Timeline**…..…………………………………………………. *p.4*

**Database Design**.…………………………………………………….*p.5*

**Client Requirements**……………………………………………….*p.6*

**Database Tables**…………………..………………………………...*p.7*

**System Functionalities**………………………………………….…*p.9*

**Admin**……………………………………………………………………..*p.13*

**Database Initialization Queries (PHP)**…………………….*p.15*

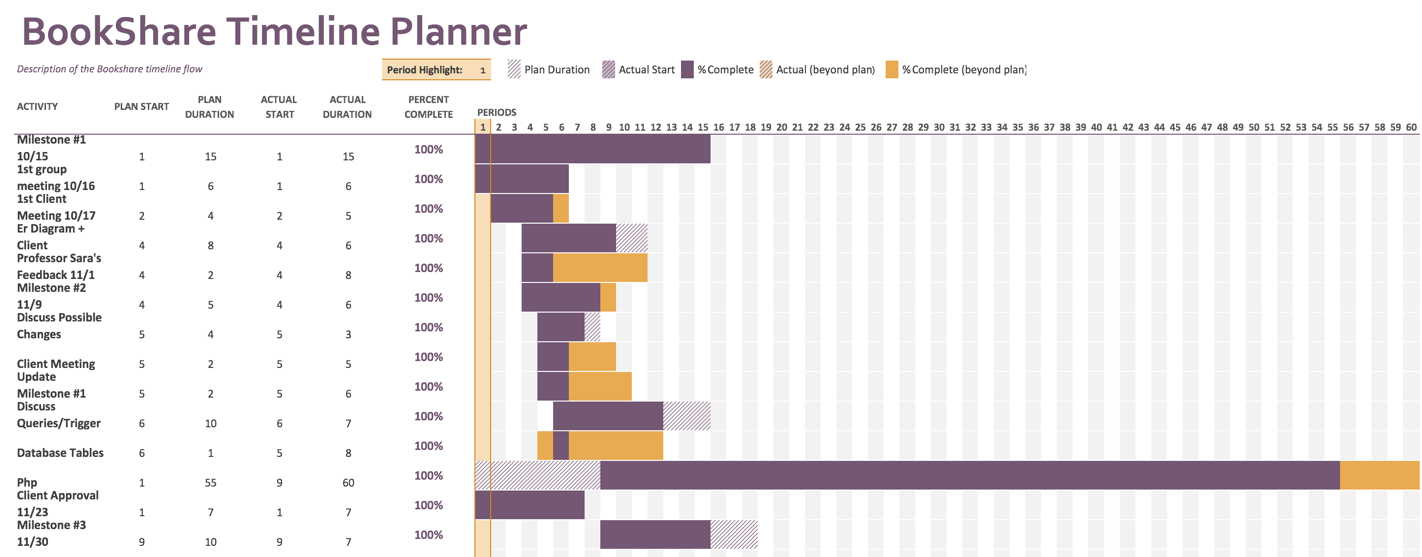
**Overview**

Our group, group #1, consisting of members Shaila Hirji, Ferdinand Tembo, Joe Do and Beverly Ackah proposes creating a web platform called BookShare where particulars can sell and purchase textbooks online.

The purpose of this project is to allow particulars to be able to buy textbooks at a lower price due to the excessive price of textbooks that many students are complaining about. Particulars will also benefit by selling the textbooks they are no longer using. Our platform aims to bring people together (Seller and Buyer).

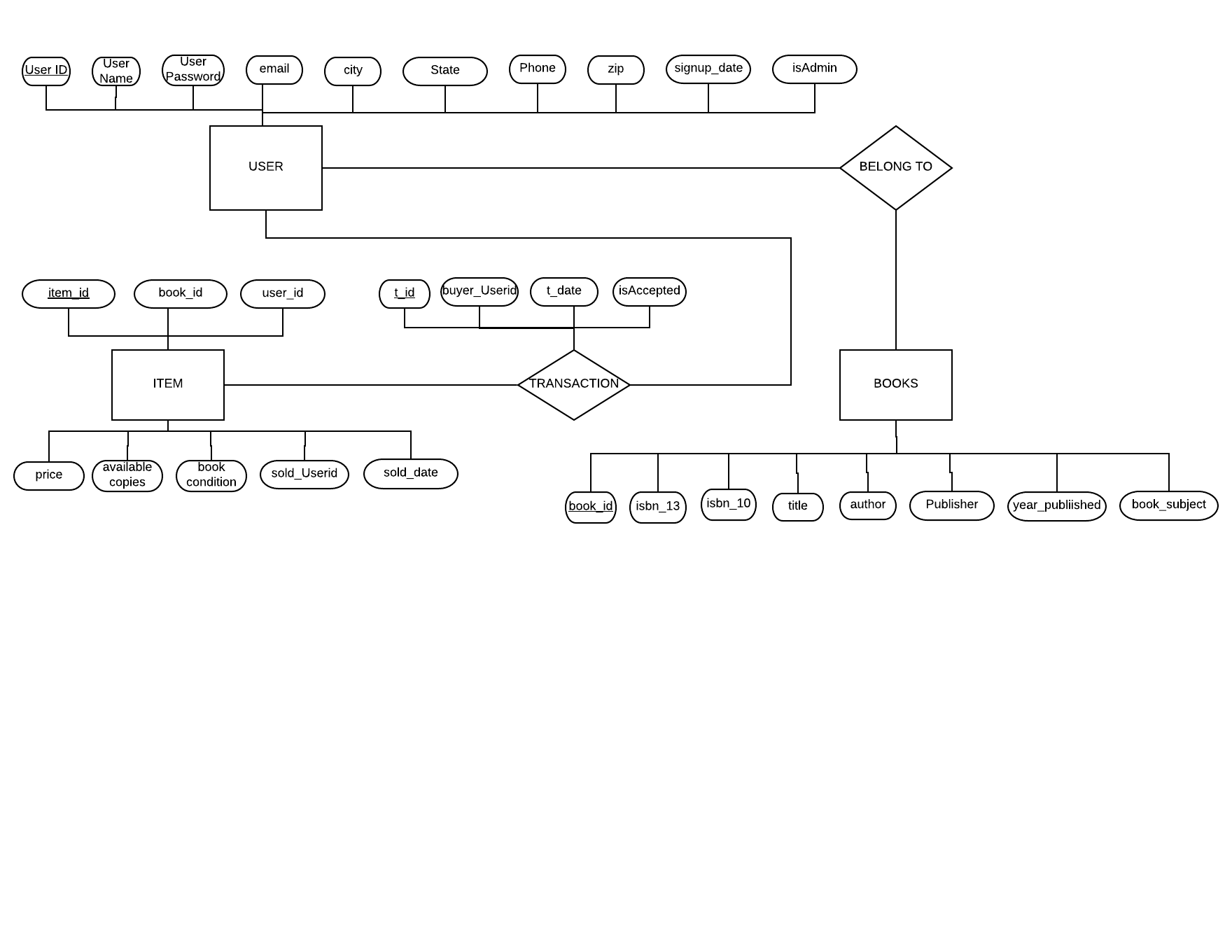
The objectives are to create a database management system that stores user credentials and textbook information such as title, ISBN, author, and date published. This backend system will allow users to input their textbooks information and to build a simple web application that allows users to query out a list of textbooks that are relevant to them and send in a request to purchase the items.

**Project Timeline**



**Database Design**

The following diagram provides a high-level introduction to our database. It is provided as reference while learning about the system’s client functionality in the following section.



**Client Requirements**

Our client, Nick Twite, wanted the following initial requirements for BookShare to run:

A platform offers the following services to the interested parties i.e. buyers and sellers who will have to create a “BookShare” account by registering in our system and then login and logout as they please. If a user just wants to browse through the system and see available book, they are free to do so without an account. As soon as you create a BookShare account, you can have the functionalities of both a seller and a buyer.

As a seller, you are able to add a book, remove a book if its already sold, or if the seller is no longer interested in selling the book. see which books you are currently selling, see the requested books from different potential buyers and decide whether you want to accept them or leave them in “pending” status.

As a buyer, you are able to search a book of your interest, send requests to buy a book from a particular seller. A buyer will be able to access the seller’s information and access the status of the request of the books they might want to purchase from a seller. A buyer will be able to access the list of the items that he/she has bought on BookShare.

The system is set to have only one admin for the moment. The admin has the functionalities of a seller and a user. In addition to that, the admin will be able to keep track of its participants as well as their date of registration. And the total sales made within a specific period of time.

**Database Tables**

This section describes what data we will store in our database as well as how our relations represent this data. The following tables will be included in our database. Please refer to the E-R diagrams at the beginning of the document for a higher-level graphical description.

**User**

User (user\_id INT AUTO\_INCREMENT NOT NULL, user\_Name VARCHAR(225), user\_Password VARCHAR(50), email VARCHAR(80), city VARCHAR(30), state CHAR(5), phone VARCHAR(15), zip INT, signup\_date DATETIME, isAdmin INT, PRIMARY KEY (user\_id));

Foreign key: none

Candidate Key: user\_id

Primary Key: user\_id

Not Null: user\_id

This table stores all the information associated with a given user. It is used to authenticate users attempting to login as a user. A user’s user\_id is used by other tables in order to determine which item(s) belong to each user.

**Books**

Books (book\_id INT AUTO\_INCREMENT NOT NULL, isbn\_13 VARCHAR(13) UNIQUE, isbn\_10 VARCHAR(10) UNIQUE, title VARCHAR(225) NOT NULL, author VARCHAR(225), publisher VARCHAR(225), year\_published INT, book\_subject VARCHAR(100), PRIMARY KEY (book\_id));

Foreign key: none

Candidate Key: book\_id

Primary Key: book\_id

Not Null: book\_id, title

Each entry in this table represents books available for purchase by our platform. Each book has a reference an isbn13 and isbn10, a title, an author, a publisher, its year and its subject.

**Item**

item (item\_id INT AUTO\_INCREMENT NOT NULL, book\_id INT NOT NULL, user\_id INT NOT NULL, price DECIMAL(4 , 2 ) NOT NULL, available\_copies INT, book\_condition VARCHAR(100), sold\_UserId int, sold\_date datetime, PRIMARY KEY (item\_id), FOREIGN KEY (book\_id) REFERENCES books (book\_id), FOREIGN KEY (user\_id) REFERENCES users (user\_id), FOREIGN KEY (sold\_userId) REFERENCES users (user\_id));

Foreign key: book\_id, user\_id, sold\_userId

Candidate Key: item\_id

Primary Key: item\_id

Not Null: item\_id, book\_id, user\_id, price

The item table manages the inventory. For each item, there is a transaction date associated, the number of copies, its status, the item condition and its price.

**Transaction\_Table**

CREATE TABLE transaction\_table (t\_id INT AUTO\_INCREMENT NOT NULL, item\_id INT NOT NULL, buyer\_Userid INT NOT NULL, t\_date DATETIME, isAccepted boolean, PRIMARY KEY (t\_id), FOREIGN KEY (item\_id) REFERENCES item (item\_id), FOREIGN KEY (buyer\_Userid) REFERENCES users (user\_id));

Foreign key: item\_id, buyer\_Userid

Candidate Key: t\_id

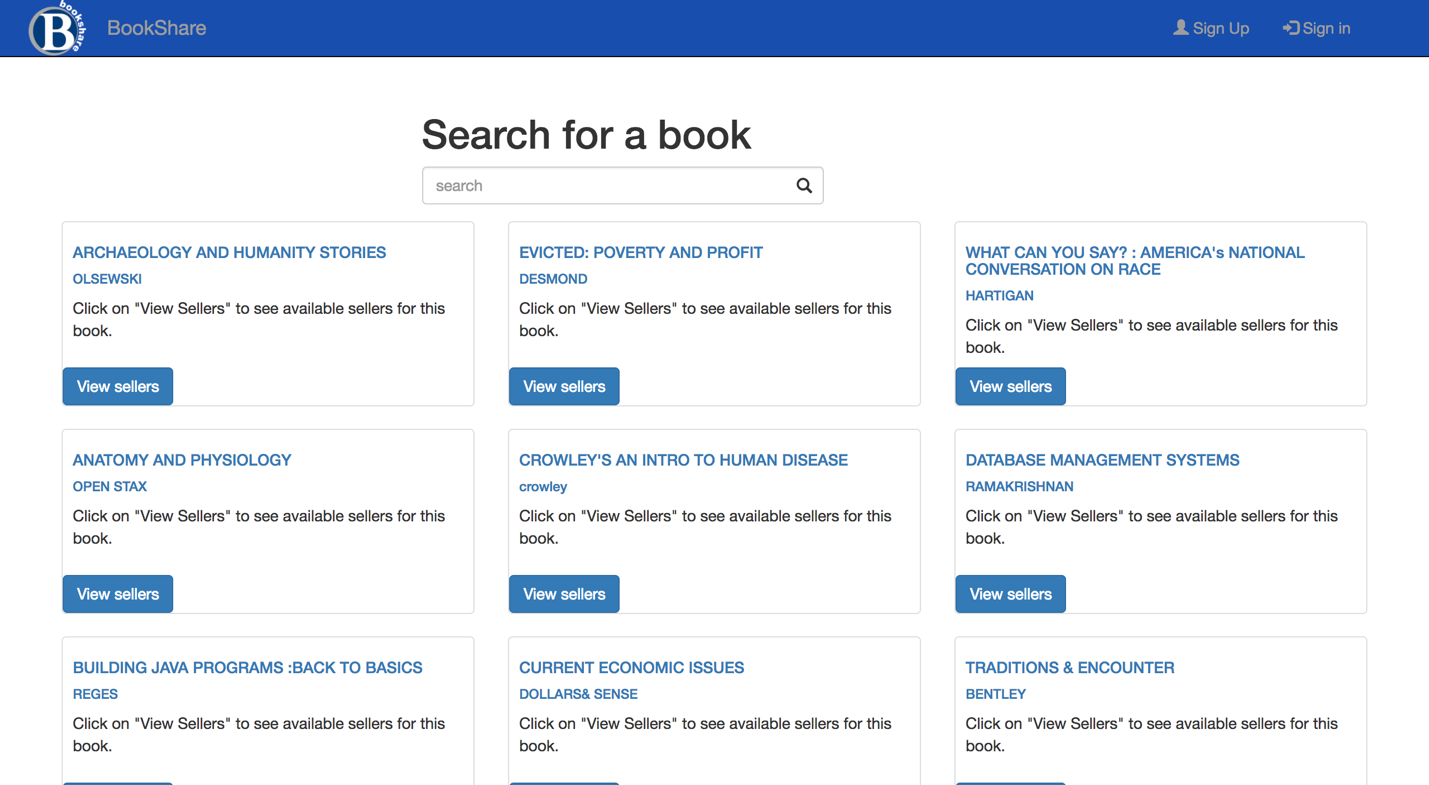
Primary Key: t\_id

Not Null: t\_id, buyer\_UserId

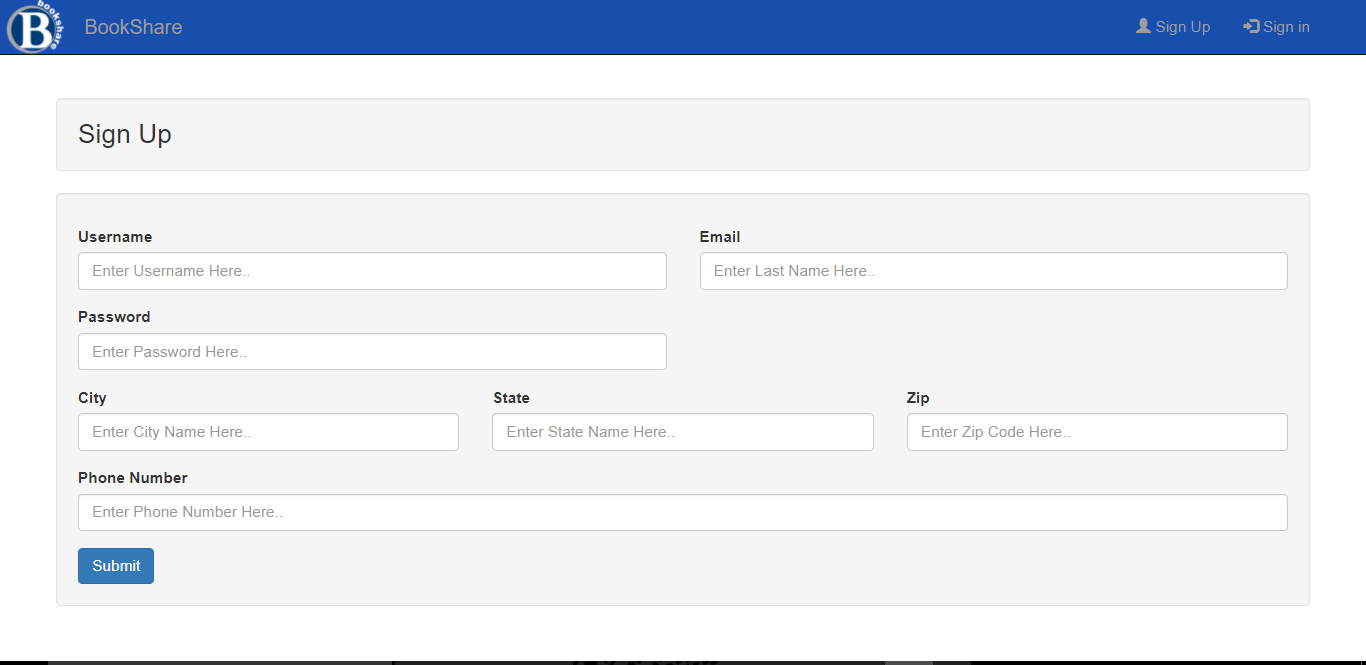
This table records the transaction between a user and an item. It is used to keep track of the date of the transaction as well as its status.

**System Functionalities**

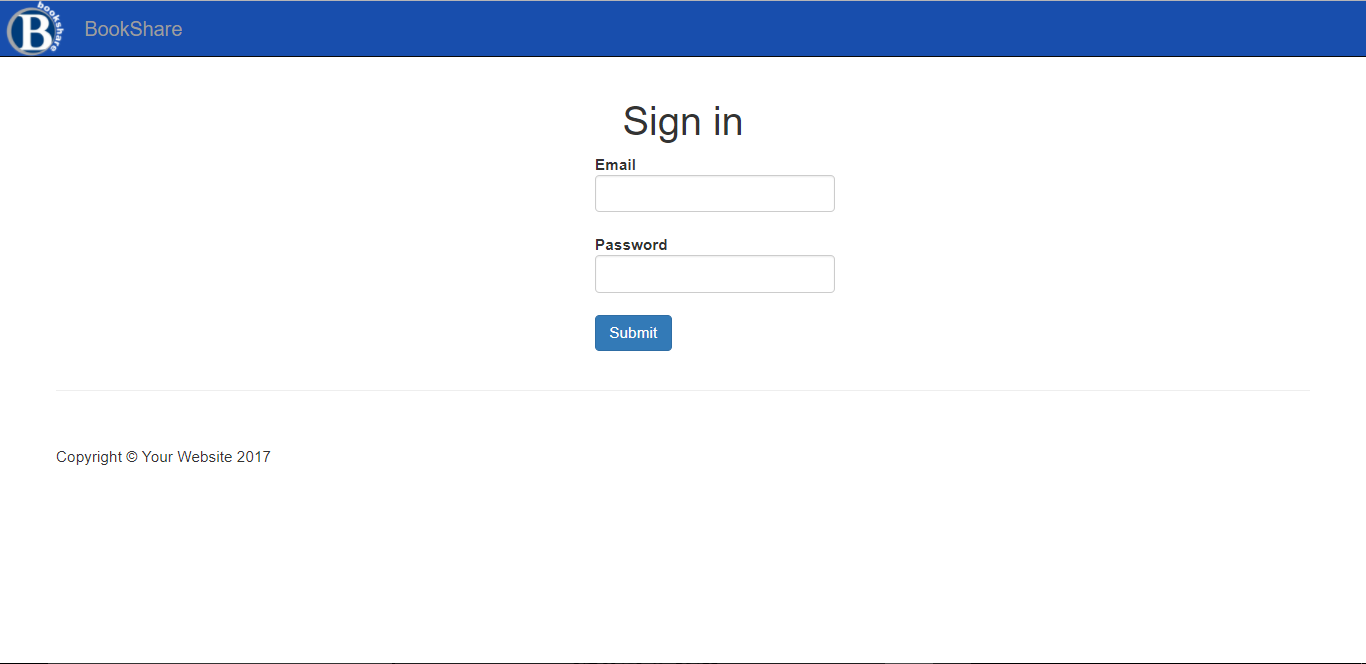
Anyone can view all the available books without signing up.



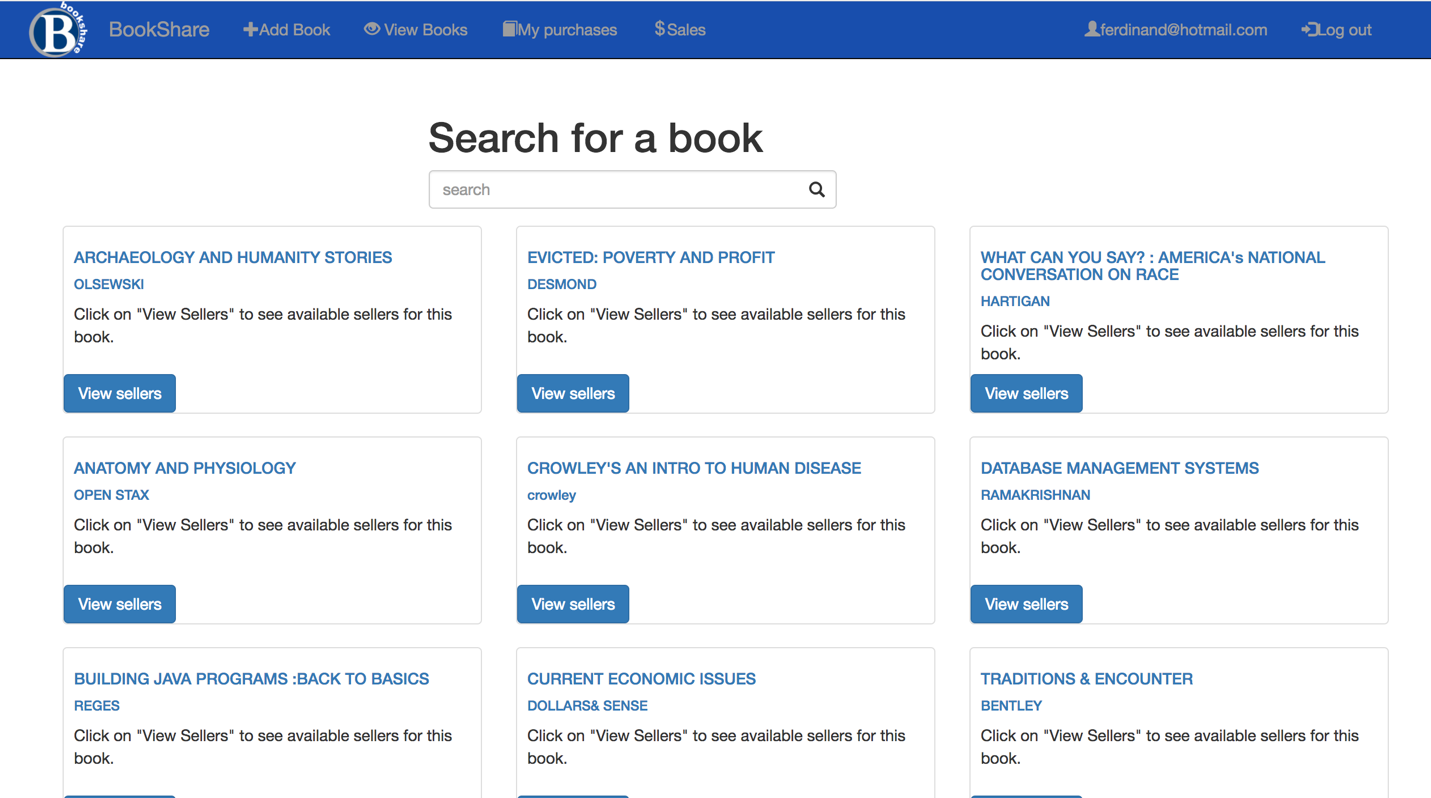
If you are interested in buying a book, then you will have to sign up by clicking the right top corner Sign Up button. You will have to enter the following information: a username, email address, password, city, state, zip code and phone number. Click the Submit button to create your BookShare account.



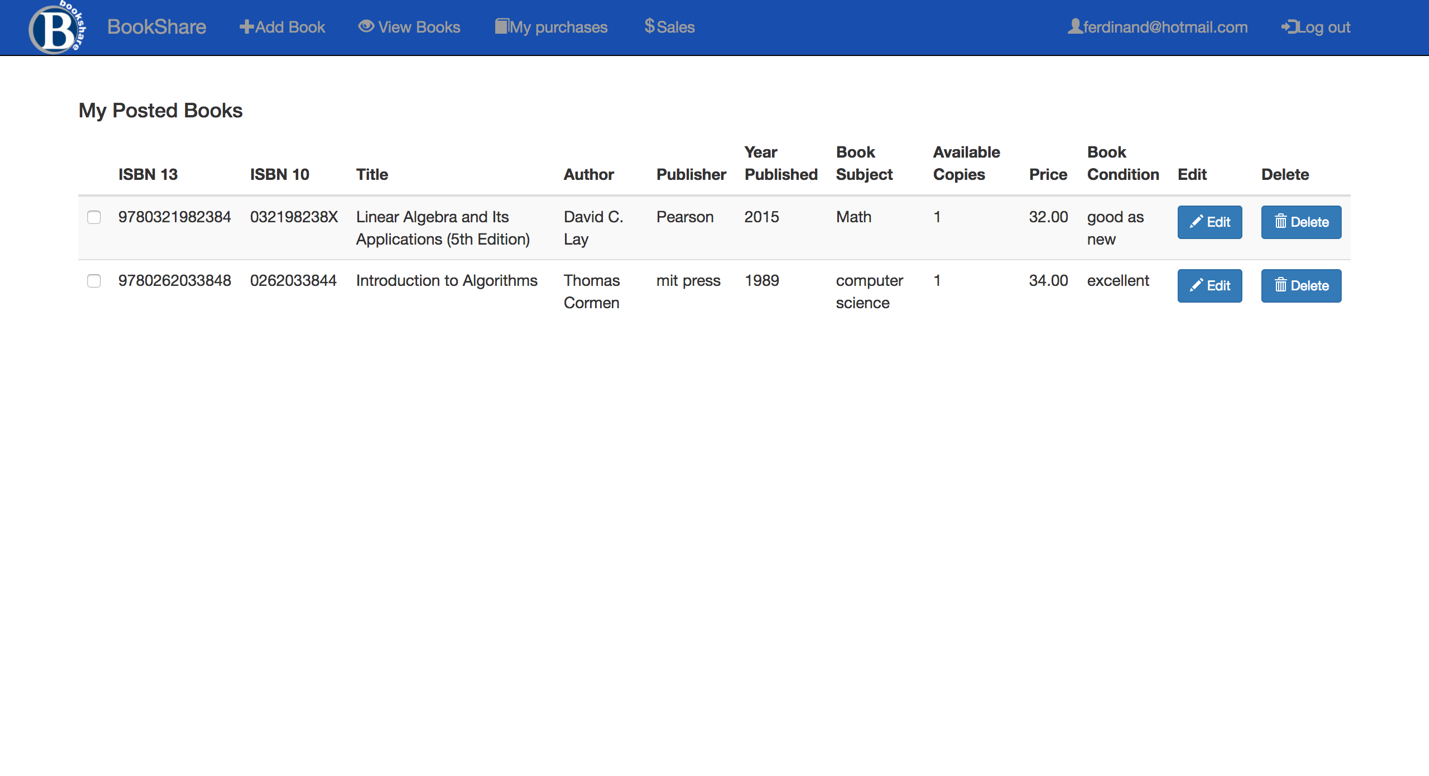
You will automatically be signed in the system and ready to access the different features of BookShare. For future use of the BookShare system, you will be able to sign in by entering your email address and password in the Sign In page.



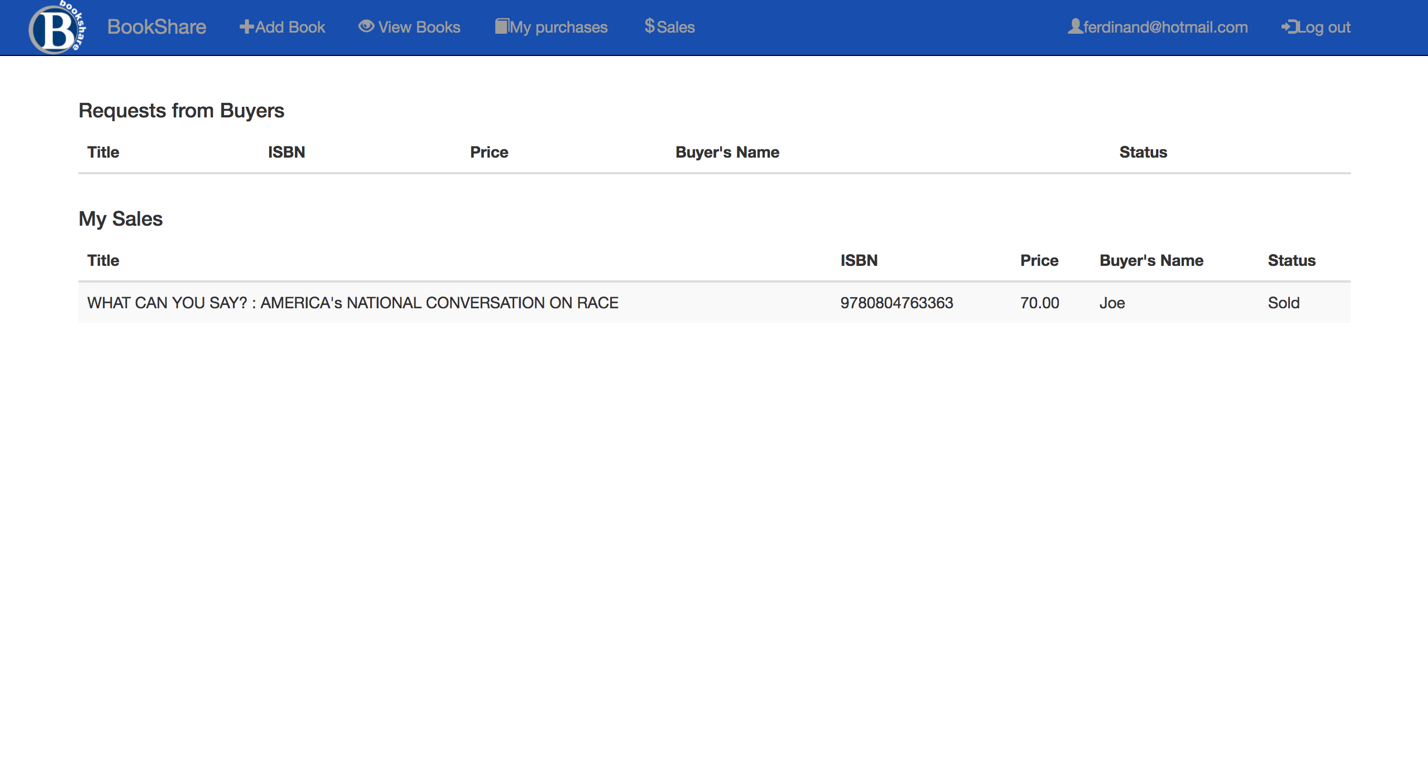
Now that you have a BookShare account, you are able to access more features of the system such as adding a book by doing the following:



You can view all the unsold books you have under your account by clicking the View Books Tab. Views books has many functionalities such as Editing if there has been a typo mistake or Deleting the book you no longer want to sell it.



You will be able to see the sales you have made once you sell one of your books

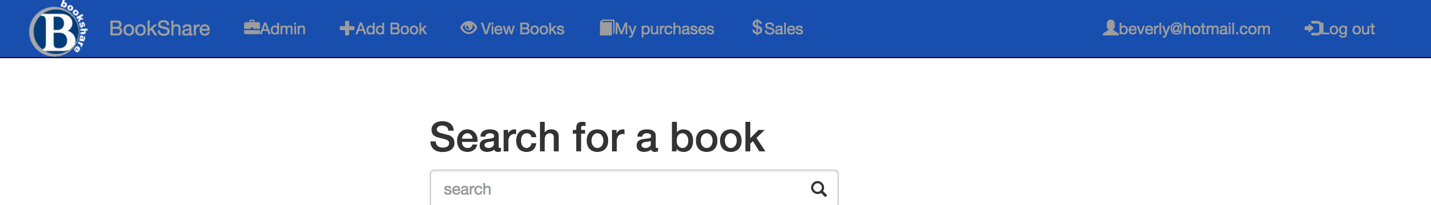


As a buyer, you will be able to see the items you have purchased on BookShare.

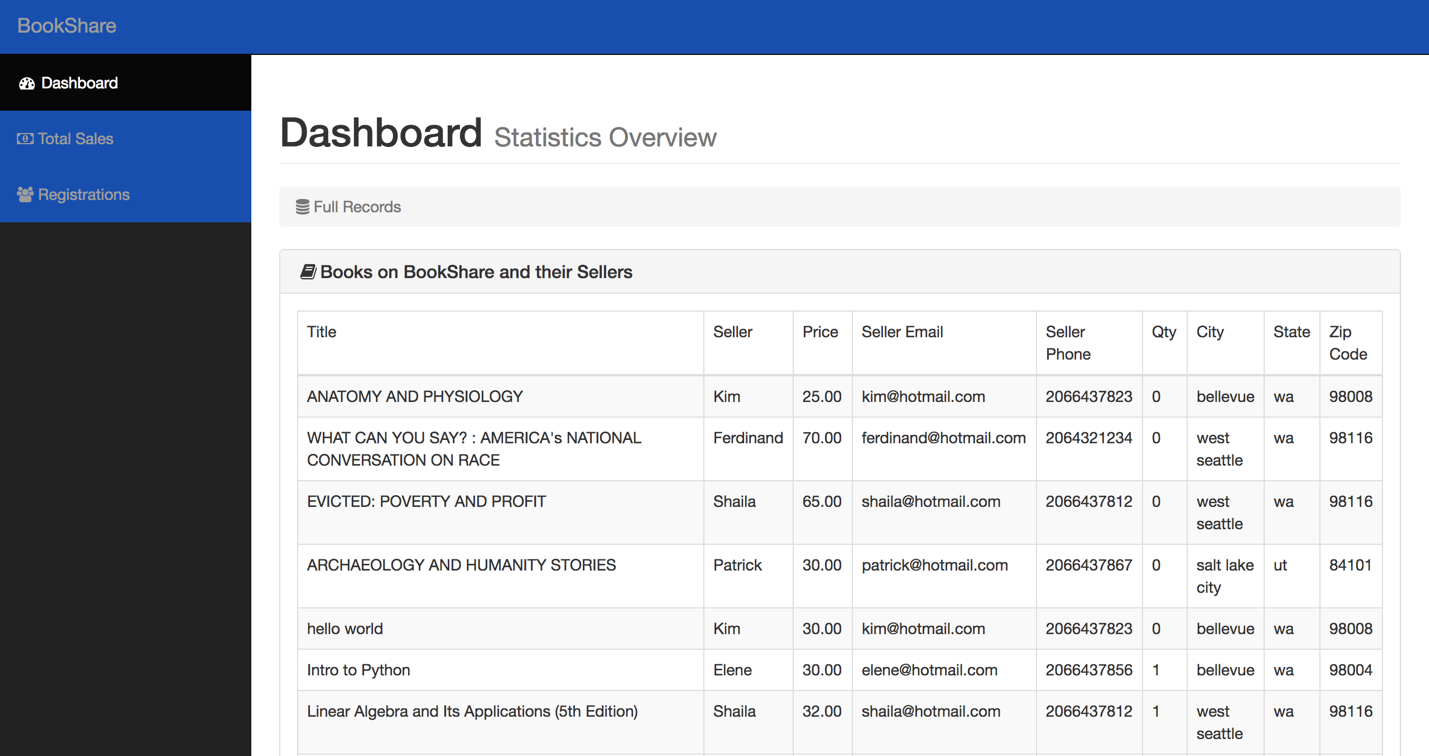


**Admin**

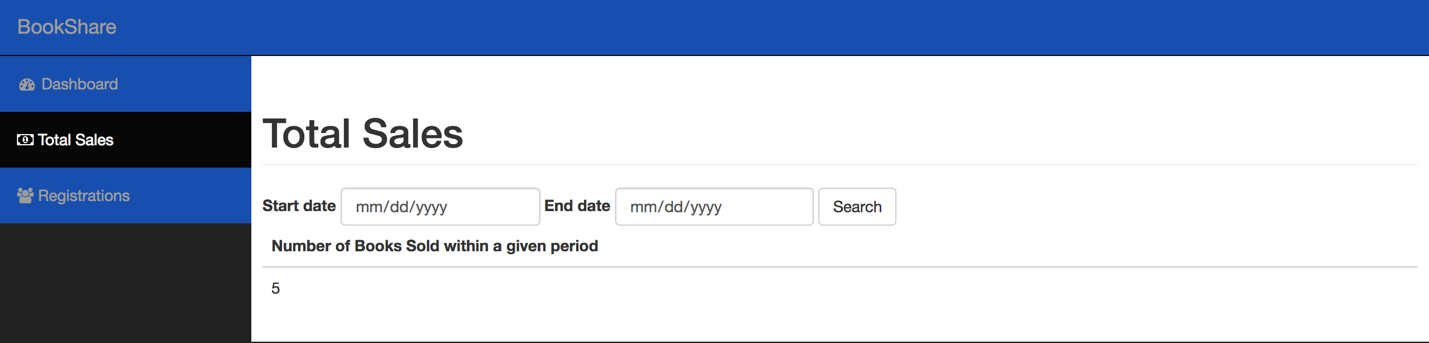
BookShare can only have one admin who has access to the Admin tab.



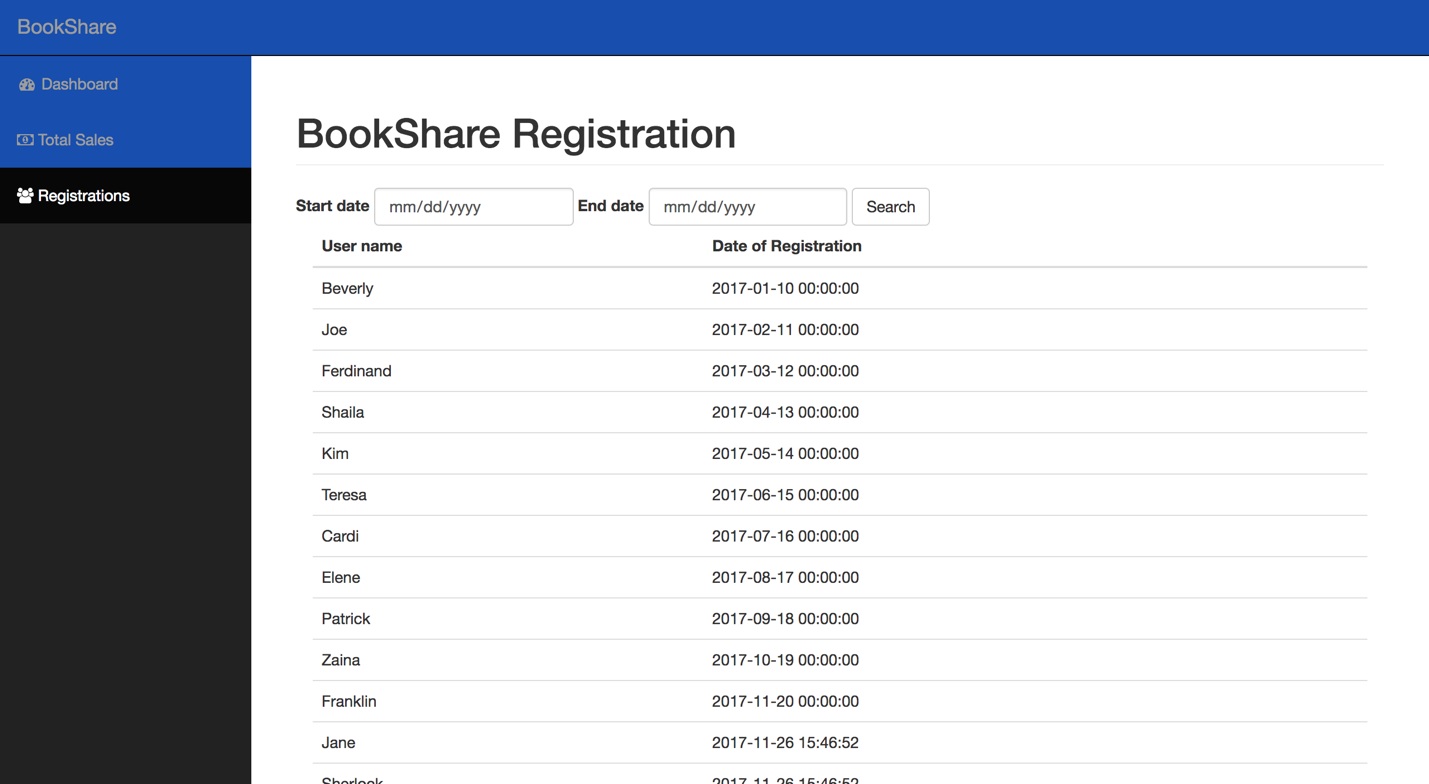
An admin has access to the information of the sellers using the Dashboard Tab.



The admin can access the total sales within a given period by using the Total Sales tab.



The admin can see the number of registrations on BookShare.



**Database Initialization Queries (PHP)**

Sign Up

$string = "insert into BookShare.users (user\_Name,email,user\_Password,city, state, phone, zip, signup\_date, IsAdmin)

values ('{$username}','{$email}','{$password}','{$city}' , '{$state}' , '{$zip}' , '{$phone}',now(), 0)";

$result = query($string)’

Sign In

$query = query("select email , user\_id from BookShare.users where email='{$email}' AND user\_Password='{$password}'");

$row = fetch\_array($query);

$user\_email = $row['email'];

Add book if it does not exist in book table

$querystep1 = "insert into BookShare.books (isbn\_13,isbn\_10,title,author,publisher,year\_published,book\_subject)

values('{$isbn13}', '{$isbn10}','{$title}','{$author}','{$publisher}','{$year\_published}','{$book\_subject}');

" ;

$insert\_book = query($querystep1);

if(!$insert\_book){

set\_message("Your book was not added! :( ");

redirect("addbook.php");

}

Insert into item if it exists

$querystep2 = "select book\_id from BookShare.books where books.isbn\_13='{$isbn13}'";

$book\_id = fetch\_array(query($querystep2))['book\_id'];

$user\_id = $\_SESSION['user\_id'];

$querystep3 = "insert into BookShare.item(book\_id,user\_id,price,available\_copies,book\_condition)

values ('{$book\_id}','{$user\_id}','{$price}','1','{$book\_condition}')";

$insert\_item = query($querystep3);

Homepage Show book

$query = query("SELECT \* FROM BookShare.books");

Search book

$searchString = trim($\_GET['search']); //

$queryString = "select book\_id, title , author,isbn\_13 ";

$queryString .= "from BookShare.books where isbn\_13='{$searchString}' OR isbn\_10='{$searchString}' ";

$queryString .= "OR title like CONCAT('%', '{$searchString}', '%') or author like ";

$queryString .= "CONCAT('%', '{$searchString}', '%') or publisher like ";

$queryString .= "CONCAT('%', '{$searchString}', '%') or year\_published='{$searchString}' ";

$queryString .= "or book\_subject like CONCAT('%', '{$searchString}', '%')"; //

View Sellers

$searchString = trim($\_GET['search']);

$queryString = "select book\_id, title , author,isbn\_13 ";

$queryString .= "from BookShare.books where isbn\_13='{$searchString}' OR isbn\_10='{$searchString}' ";

$queryString .= "OR title like CONCAT('%', '{$searchString}', '%') or author like ";

$queryString .= "CONCAT('%', '{$searchString}', '%') or publisher like ";

$queryString .= "CONCAT('%', '{$searchString}', '%') or year\_published='{$searchString}' ";

$queryString .= "or book\_subject like CONCAT('%', '{$searchString}', '%')";

Request purchase

$queryString = "insert into BookShare.transaction\_table (item\_id,buyer\_Userid,t\_date,isAccepted)

values('{$item\_id}','{$user\_id}',now(),0)";

$query = query($queryString);

Display request

$queryString = "select distinct b.title as Title,b.isbn\_13 as 'ISBN 13', i.price as Price, t.t\_date as 'Date of Transaction', t.isAccepted as 'status',u.user\_name as 'Seller Name'

from BookShare.books b INNER JOIN BookShare.item i on b.book\_id=i.book\_id

inner join BookShare.transaction\_table t on i.item\_id=t.item\_id

inner join BookShare.users u on u.user\_id = i.user\_id

where t.buyer\_Userid='{$user\_id}'";

View books from your account

$queryString = "select distinct b.isbn\_13,i.item\_id, i.book\_id,b.isbn\_10,b.title,b.author,b.publisher,b.year\_published,b.book\_subject,i.available\_copies,i.price,i.book\_condition

from BookShare.books b INNER JOIN BookShare.item i on b.book\_id=i.book\_id

where i.user\_id='{$userid}' AND i.available\_copies>0";

$query = query($queryString);

$rows = mysqli\_num\_rows($query);

Get an update of the book

$queryString = "select \* from BookShare.books where book\_id ='{$id}'";

Delete book

$queryString = "update BookShare.item i set i.available\_copies=0 where i.item\_id='{$itemid}'";

View requests

$queryString = "select b.title as Title,i.item\_id,b.isbn\_13 as 'ISBN\_13', i.price as Price, t.buyer\_UserId, buyer\_request.user\_name as 'Requested\_by'

from BookShare.books b INNER JOIN BookShare.item i on b.book\_id=i.book\_id

left join BookShare.users u on u.user\_id = i.user\_id

left join BookShare.transaction\_table t on t.item\_id=i.item\_id

left join BookShare.users buyer\_request on buyer\_request.user\_id = t.buyer\_userId

where i.user\_id='{$userid}' and i.available\_copies =1 and t.isAccepted=0";

View Sales

$queryString = "select b.title as Title,i.item\_id,b.isbn\_13 as 'ISBN\_13', i.price as Price, t.buyer\_UserId, buyer\_request.user\_name as 'Requested\_by'

from BookShare.books b INNER JOIN BookShare.item i on b.book\_id=i.book\_id

left join BookShare.users u on u.user\_id = i.user\_id

left join BookShare.transaction\_table t on t.item\_id=i.item\_id

left join BookShare.users buyer\_request on buyer\_request.user\_id = t.buyer\_userId

where i.user\_id='{$userid}' and t.isAccepted=1";

Approve Sales

$queryString1="update BookShare.item i set i.available\_copies=0,sold\_UserId ='{$buyerid}', sold\_date=now()

where i.item\_id = '{$itemid}'";

$query1 = query($queryString1);

if($query1){

$queryString2="update BookShare.item i set i.sold\_UserId='{$buyerid}'

where i.item\_id='{$itemid}'";

$query2 = query( $queryString2);

if($query2){

$queryString3="update BookShare.transaction\_table t set isAccepted=1

where item\_id='{$itemid}' and buyer\_Userid={$buyerid}";

$query3 = query( $queryString3);

if($query3){

$queryString4 = "update BookShare.transaction\_table t set isAccepted=2

where item\_id='{$itemid}' and buyer\_Userid<>{$buyerid}";

Show users (admin)

$stringQuery = "SELECT \* FROM BookShare.users";

Show partial users

$stringQuery = "SELECT user\_name AS 'Name', signup\_date AS 'Date\_of\_Sign up'

FROM BookShare.users WHERE signup\_date BETWEEN '{$startdate}' AND '{$enddate}' ORDER BY signup\_date DESC";

Show books (admin)

$stringQuery = "select \* from BookShare.item i

where i.sold\_UserId is NOT null";

Show partial books (admin)

$stringQuery = "select \* from BookShare.item i

where i.sold\_UserId is NOT null and i.sold\_date between '{$startdate}' and '{$enddate}' ";

All books

$queryString = "select title as 'Title', user\_Name as 'Seller',price as 'Price', email as 'Seller\_Email', phone as 'Seller\_Phone', available\_copies as 'Qty',city as 'City', state as 'State', zip as 'Zip\_Code'

from BookShare.books b inner join BookShare.users u inner join BookShare.item i ON u.user\_id=i.user\_id and i.book\_id=b.book\_id";