----------

Student 1

Name: Sayyid Iskandar Khan

Student number: 6959751

Email address: sisoa078@uowmail.edu.au

Assignment number: 1

----------

**Files included in this assignment submission:**

**PHP Files – Main Files**

1) index.php - user will login and logout and main dashboard is from this page

2) config.php – manages the SQL connection & creation of databases and records & also used to store global settings

**PHP Files – Main Files  
#User Management**3) usermgt/userlist.php – shows all the user accounts (only accessible via librarian)

4) usermgt/register\_borrower.php – signup as a borrower

5) usermgt/register\_librarian.php – signup as a librarian

6) usermgt/delete.php – delete users (only accessible via librarian)

**#Resource Management – Librarian**

7) resourcemgt/librarian/resourcelist.php – shows all the resources to librarian only

8) resourcemgt/librarian/notificationbox.php – a code extension to resourcelist.php to display notification on the resourcelist page

9) resourcemgt/librarian/add.php – add new resource

10) resourcemgt/librarian/edit.php – edit resource

11) resourcemgt/librarian/delete.php – delete resource

**#Resource Management – Librarian**

12) resourcemgt/borrower/availableresourcelist.php – shows the resources applicable to borrower only

13) resourcemgt/borrower/notificationbox.php – a code extension to availableresourcelist.php to display notification on the availableresourcelist page

14) resourcemgt/borrower/borrow.php – borrow a resource

15) resourcemgt/borrower/extend.php – extend a resource

16) resourcemgt/librarian/return.php – return a resource

**Database Files**

17) database/resources.sql – SQL query used to create resources database

18) database/users.sql - SQL query used to create users database

19) database/dataset\_resourses.sql – SQL query used for dummy datasets for resources table

20) database/dataset\_users.sql – SQL query used for dummy datasets for users table

**Class File**

21) classes/resources.php - stores the resourses class info

22) classes/user.php - stores the user class info

**CSS File**

23) css/main.css - css styling common to all the pages

24) css/flex.css - css styling which helps to dynamically adjust sizing of boxes

25) css/tableui.css - css styling for tables used in the pages

26) css/notification.css - css styling for notification page only

**Page Directory**

This is all the pages directory available in the project. There is a total of 7 pages. The arrows are the navigation process to access the page and cannot be skipped. If user skips or access with an incorrect account type login, they will not be able to view the page.  
  
Also for registration of accounts, librarian can help to sign up for new librarians and new borrowers. New librarians can sign up on their own without librarian access with a pin.  
the password is only made known to other librarians. The pin is 1234 for the purposes of setting up basic security.

Index.php

resourcelist.php  
(librarian access)

add.php

edit.php

userlist.php  
(librarian access)

Register\_borrower.php  
(account registration)

Register\_librarian.php  
(account registration)

Availableresourcelist.php  
(borrower access)

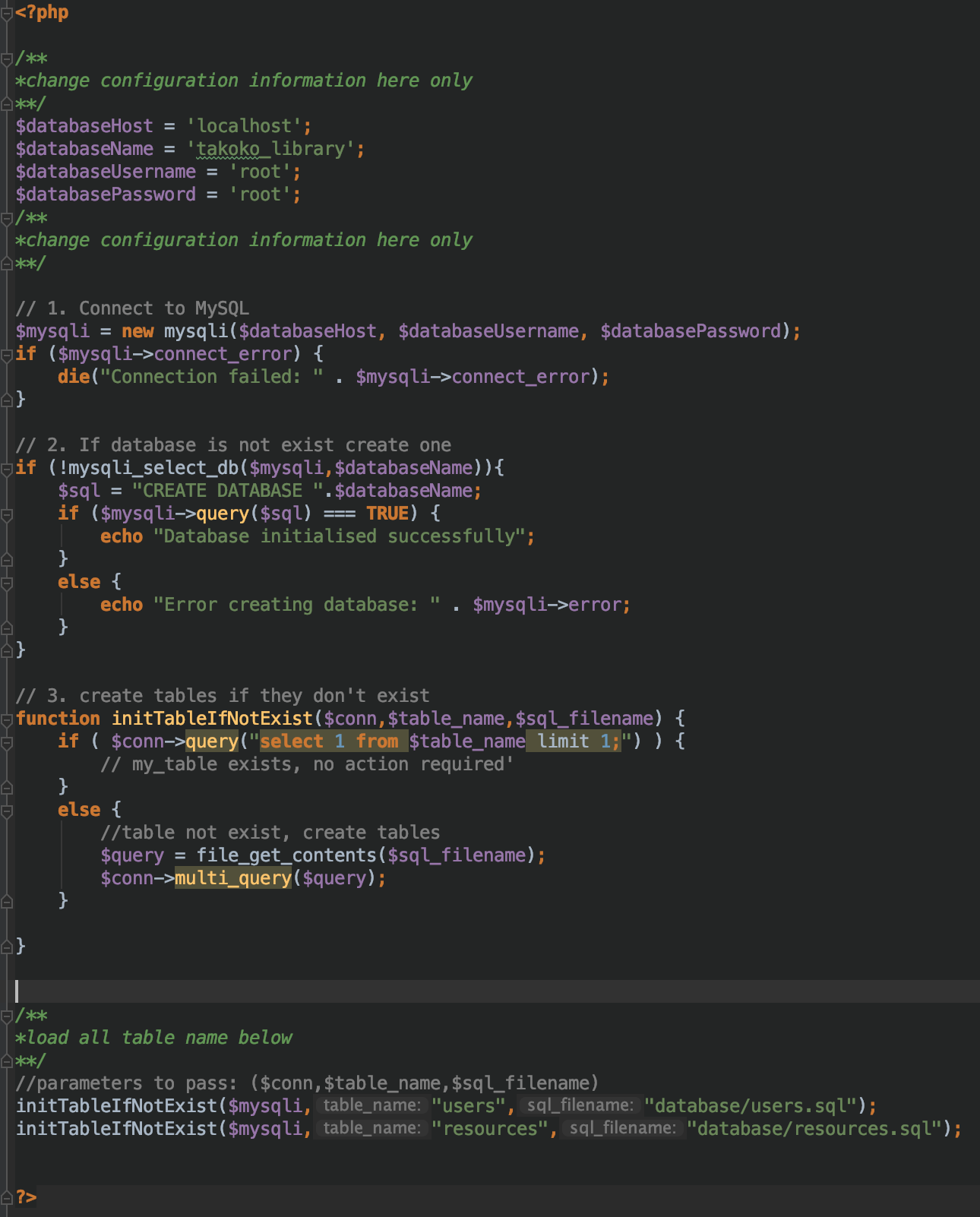
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section no** | **Page** | **Actions can be done on the page** | | | | | | | |
| 1 | Index.php | login | Logout | Display menu options |  |  |  |  |  |
| 2 | resourcelist.php | view all resources | view available resources | view borrowed  resources | view extended resources | View more info | Update resource | Delete resource | Search resource |
| 3 | add.php | insert new resources |  |  |  |  |  |  |  |
| 4 | Edit.php | Edit resources |  |  |  |  |  |  |  |
| 5 | userlist.php | View all users | Display option to register new borrower | Display option to register new librarian |  |  |  |  |  |
| 6 | Register\_borrower.php | Signup borrower |  |  |  |  |  |  |  |
| 7 | Register\_librarian.php | Signup Librarian |  |  |  |  |  |  |  |
| 8 | Availableresourcelist.php |  | view available resources | view borrowed  resources | view extended resources | View more info | Borrow resource | Update resource | delete resource |

**Description for database preparation:**

No setup Required For application and database initialization

This application, will automatically create database’s (resources and users) as soon as the application is loaded. No setup is required apart from having apache + mysql installed on your computer.

This is possible because the config file will check if database / table exist, if not exist it will create the items required. *Config.php* is also put on top of each php file, so that will not need to re-write the configuration to connect to database every time. This will be the main file used to connect to all databases across the application.



Setup is still required for data preparation

This is optional. You can walk through the entire application without requiring datasets and able to create all of the records without requiring to insert data into the database beforehand. Likewise, if u would like to try out with data prepared for you for testing purposes, u can use  
-> database/dataset\_resourses.php

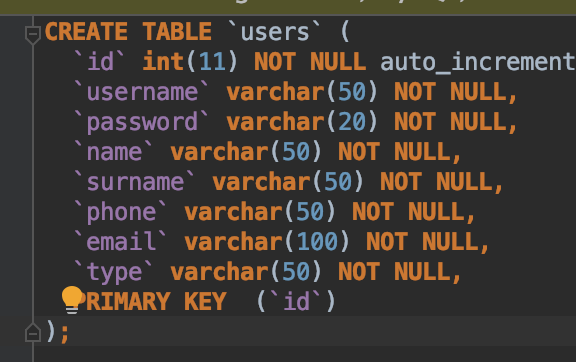
-> dataase/dataset\_users.php

To pre-load dataset to expedite testing. Navigate to the the database folder and load these two records into SQL manually.

**Description for database files:**

**#users.sql**

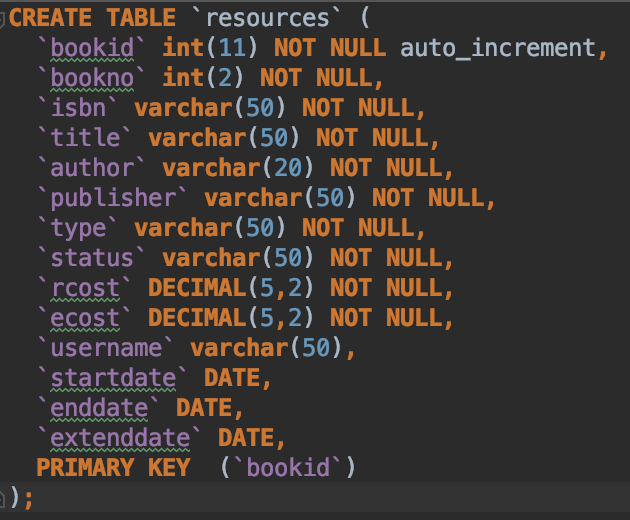
Users table is for the storage of all the users (borrowers / librarian).



|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Purpose** |
| ID | Integer | Store unique id and primary key |
| Username | String | Store username to login |
| Password | String | Store password to login |
| Name | String | Store the name of user |
| Surname | String | Store surname of user |
| Phone | String | Store phone number of user |
| Email | String | Store email of user |
| Type | String | Store type of user  (borrower / librarian) |

**#resources.sql**

Resources table is for the storage of all the resources and their status and to whom the current owner is.

****

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Purpose** |
| bookid | Integer | Store unique id and primary key |
| bookno | Integer | Store book number of the same ISBN number |
| ISBN number | String | ISBN number |
| Title | String | Title of book |
| Author | String | Book author |
| Publisher | String | Book Publisher |
| Type | String | Store type of book |
| Status | String | Specify book status  (available / borrowed / extended) |
| Rcost | DECIMAL | Regular cost of borrowing |
| Ecost | DECIMAL | Extended cost of borrowing |
| Username | String | Stores the username using the resource currently |
| Start date | DATE | Start date of borrowing |
| End date | DATE | End date of borrowing |
| Extend date | DATE | Extended date of borrowing |

**Things to note of before using the application**

Ensure you are using the correct account type to access the account otherwise u will encounter the message below stating that you have no access to the system. Unless u force navigation to the page with the wrong user access, you will not encounter this error. The reason for this is because only menu’s allowed for the user access is shown to the specific user only.

****

Sample example navigating to pages with no access with the current user access

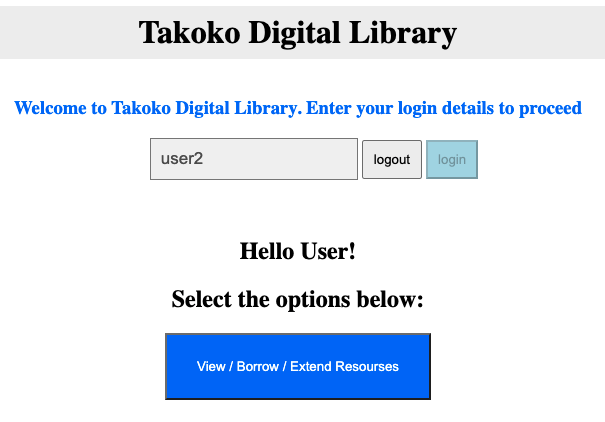
A way to verify your login access

In this application, we only have two access type,  
- Librarian  
- Borrower  
  
When you login via the admin page, you will be welcome as a  
- Librarian if user is a librarian  
- User if user is a borrower



Menu access

- Librarian (2 Menu) -> View All users | View all Resources  
- Borrower (1 Menu) -> View Available / borrow / extend Resources

****

**Description of pages for the assignment:**

**1. Index.php**The homepage of our application. Librarian and borrower will access from the same login page.

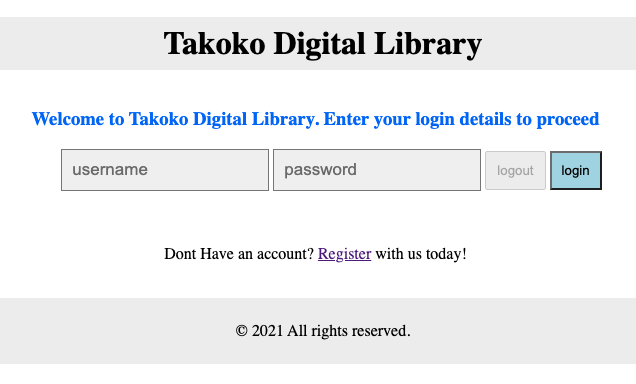
$\_SESSION[‘login’] is used to store the username upon successful login. User::init() is the use of static function within the user class to load record from database then morph it into user class object.



Logic used for the login

Before Login

Prior to login, register section and password will be accessible to user. After login, the register and password section will disappear. When $\_SESSION[‘login’] is set, these contents will be hidden from user until user login again.



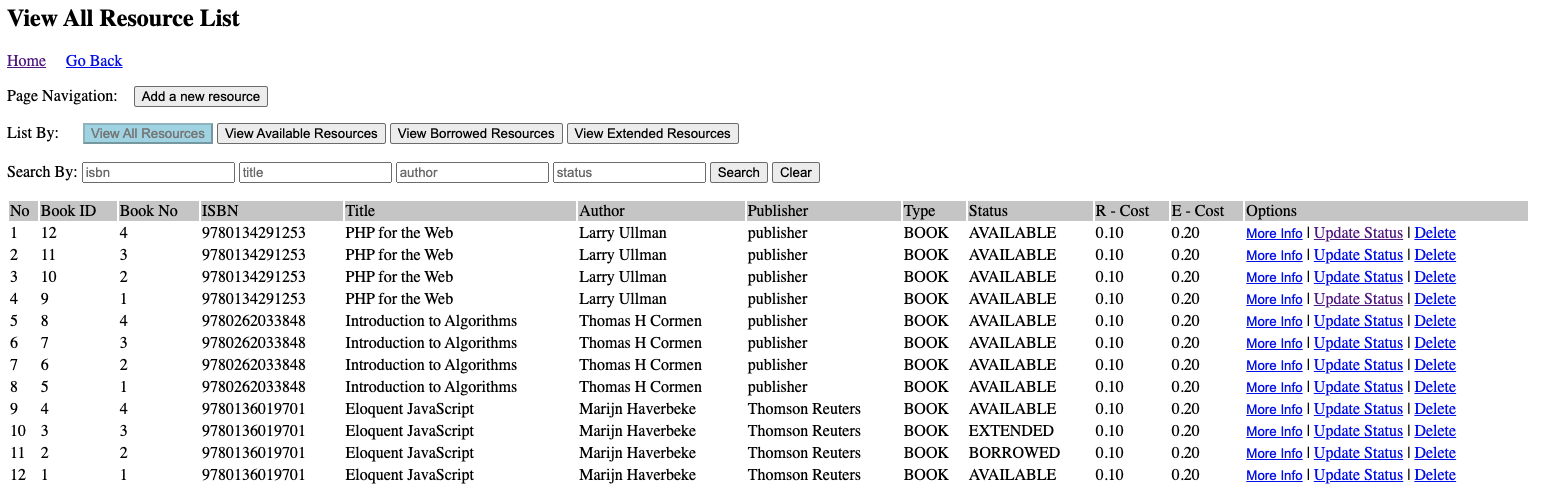
After Login

After login, username field will be disabled to prevent swapping to another user until the existing user logs out. The menu for the respective user will be shown to the user.

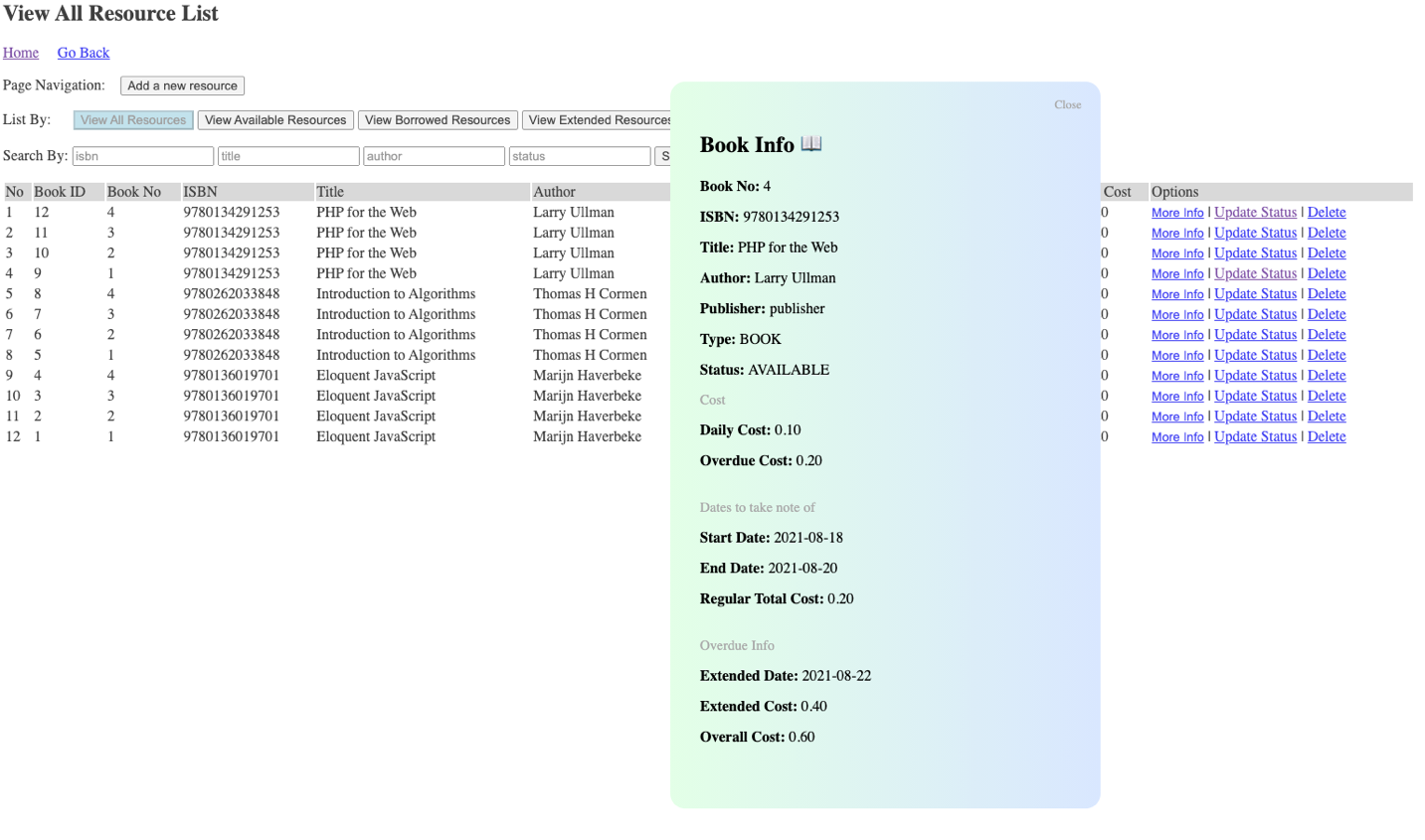


**2. resourcemgt/librarian/resourcelist.php**

this resourcelist page is to display all the resources visible to the librarian.  
Librarian can view all / available / borrowed / extended resources.  
user can click on more info to view more information via a notification screen.



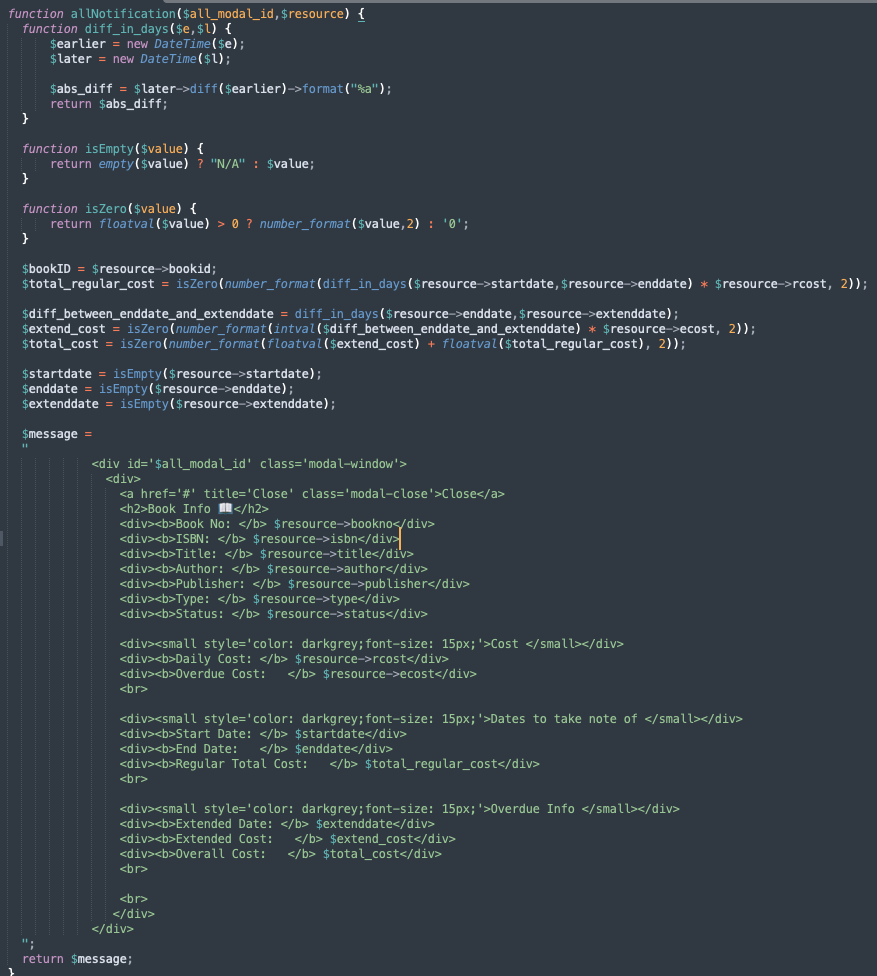
Overview of resourcelist.php



More information on the resources

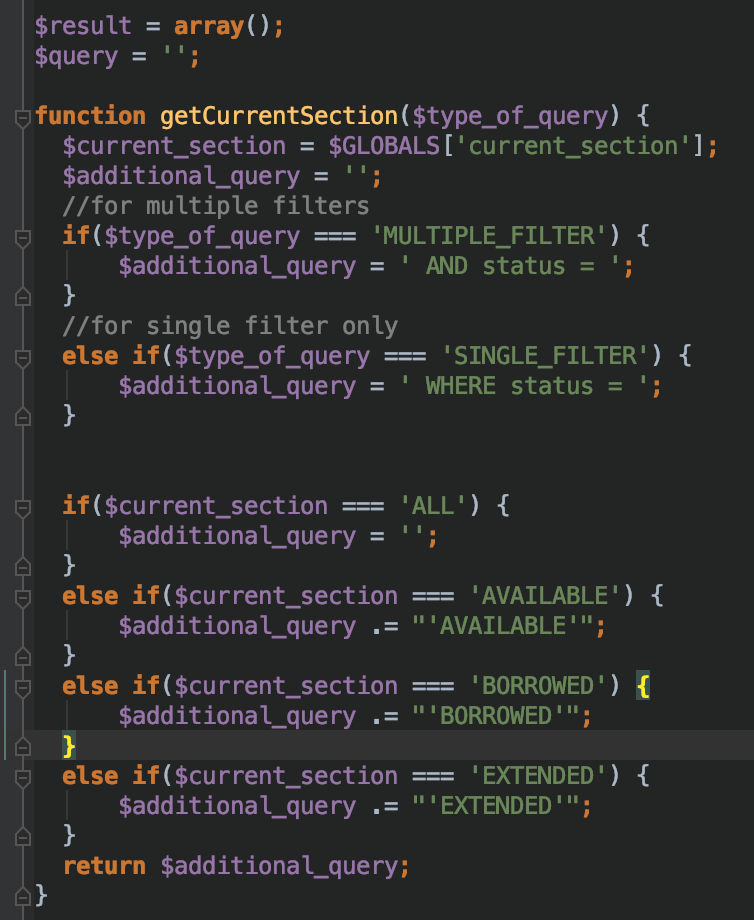
Loading of notification

Loading of the notification is via the notificationbox.php file. It is an extension file and does not have any HTML codes inside while its job is to support the rendering of the notification when the more info button is clicked. The code is separated to help make the code more manageable across the application.

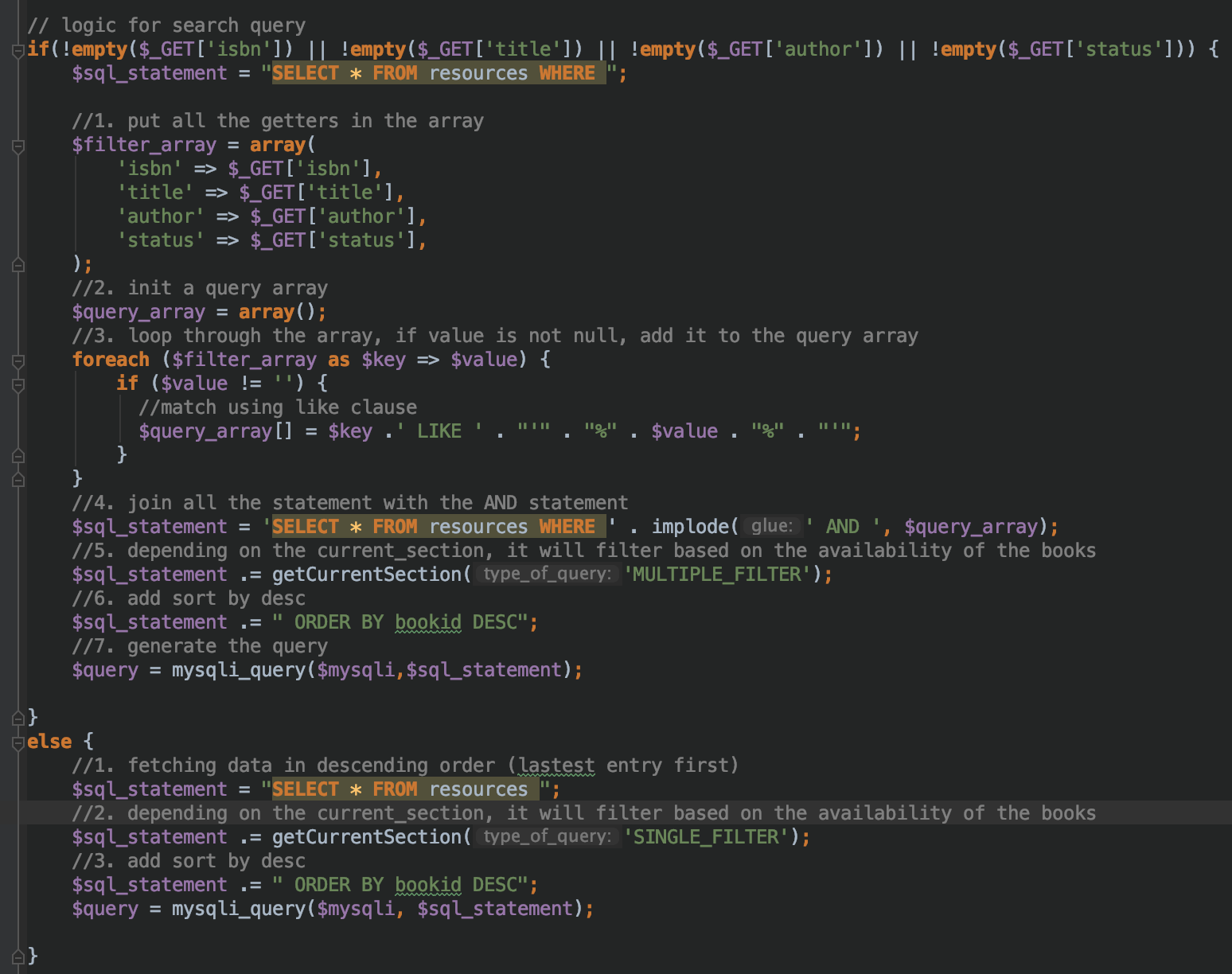


Loading of the records in this page

Loading of the resources is retrieved from the resources table.  
This first function is to confirm if user is trying to filter using 1 filter or multiple filters.



The section of the code the main logic function for getting records for the database. Combined with the previous function, it will filter based on the $\_GET request available.

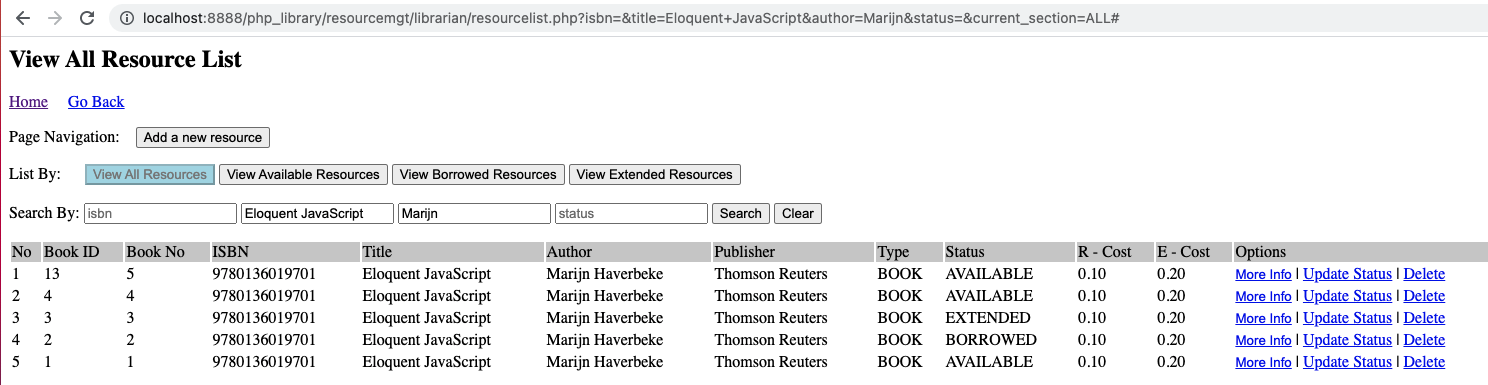


If there are records, then the records will be inserted into the resources class object. The class objects are used later to display the results on the screen row by row.

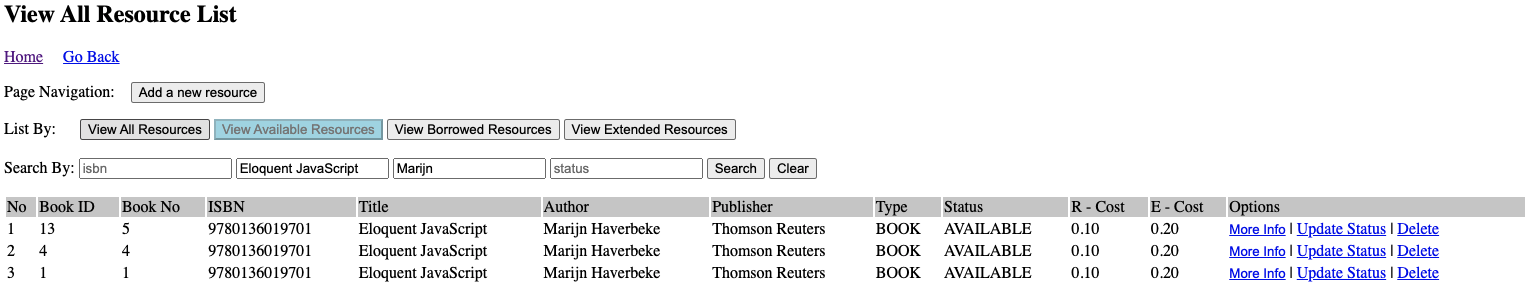


Search Records with multiple fields

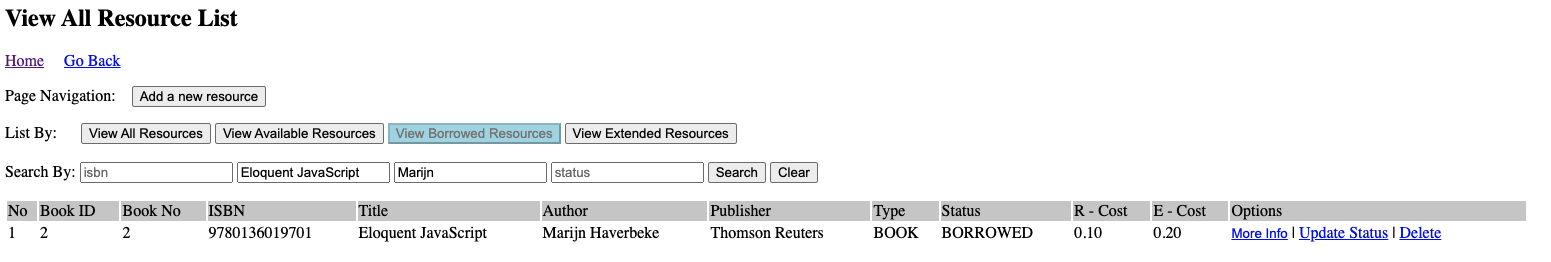
The application is able to filter 4 fields at the same time with not exact matches. This is achieved via the use of SQL LIKE clause, AND clause and checking getter parameters whether they are empty or not. This is achieved using the $\_GET request and the logic stated above.



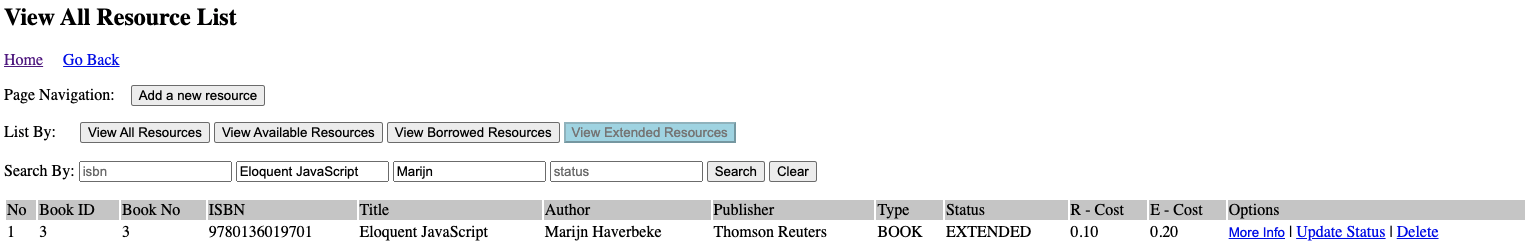
Resourcelist.php - View All Resources

****

Resourcelist.php - View Available Resources

****

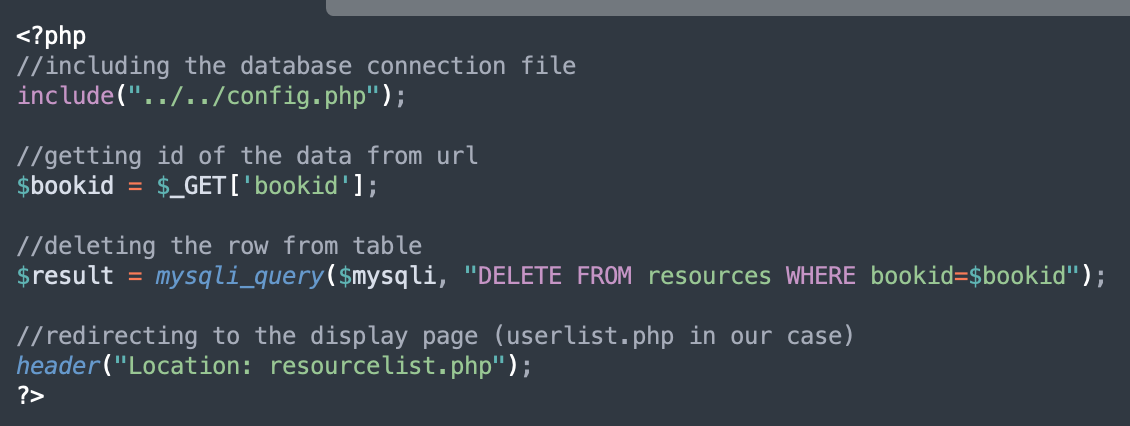
Resourcelist.php - View Borrowed Resources

****

Resourcelist.php - View Extended Resources

resourcemgt/librarian/delete.php

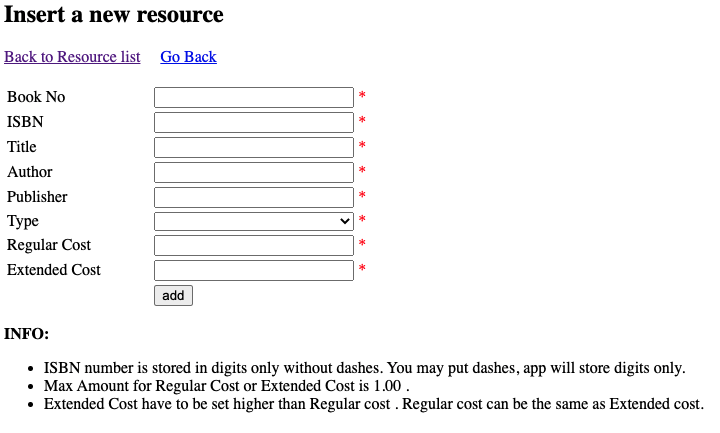
the delete.php file is not a html file but just an extension file to the resourcelist.php file so that to make the code more modular and easier to manage the logic. There is no page to visit. Once the URL is called, the resources will be deleted.



Resourcemgt/librarian/delete.php

**3. resourcemgt/librarian/add.php**

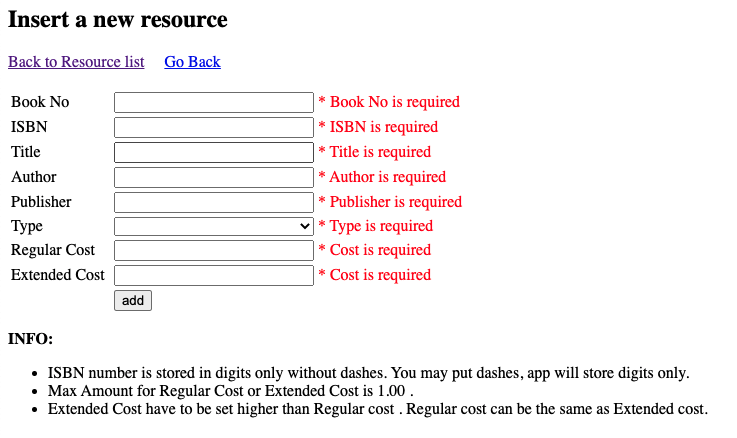
For the adding of new record, book no & ISBN combined together must be unique, otherwise will not allow insert into database. There will still be a book id created in the background to uniquely track and update or delete the book easily later. Also, there is validation to prevent insertion of blank records and fields. Records are inserted using the $\_POST request to store a huge collection of data.



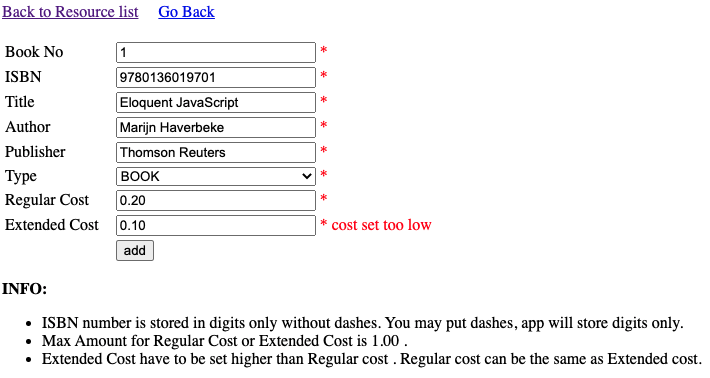
Insert new resource page when visited



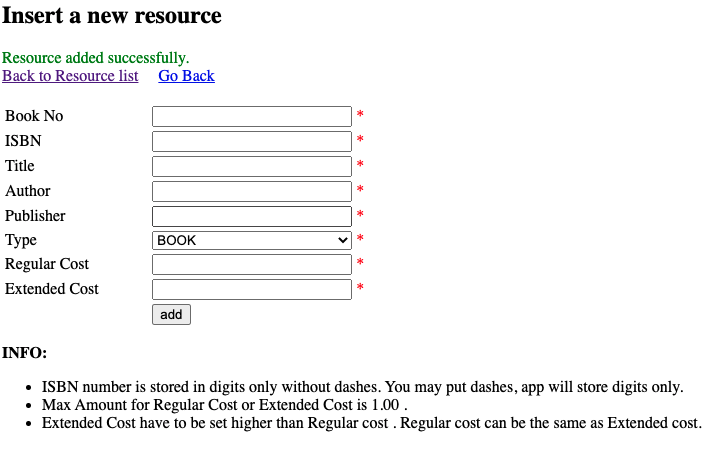
Error when user trying to insert book and ISBN number that have existed



Validation added to prevent incorrect data inserted



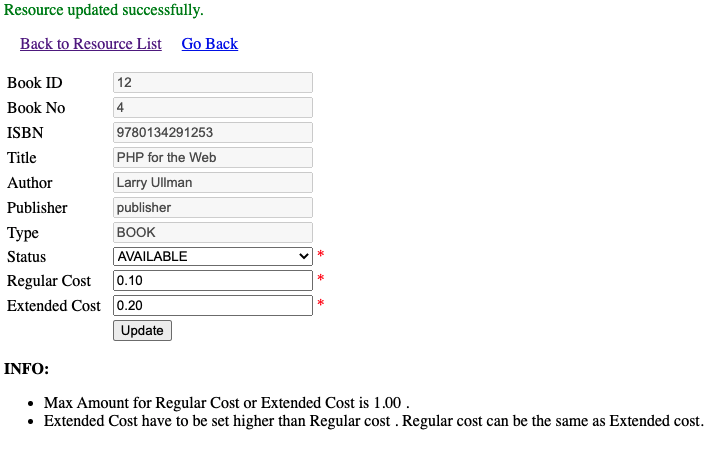
Validation to prevent regular cost greater than extended cost



successful insert on insert a new resource page

**4. resourcemgt/librarian/edit.php**

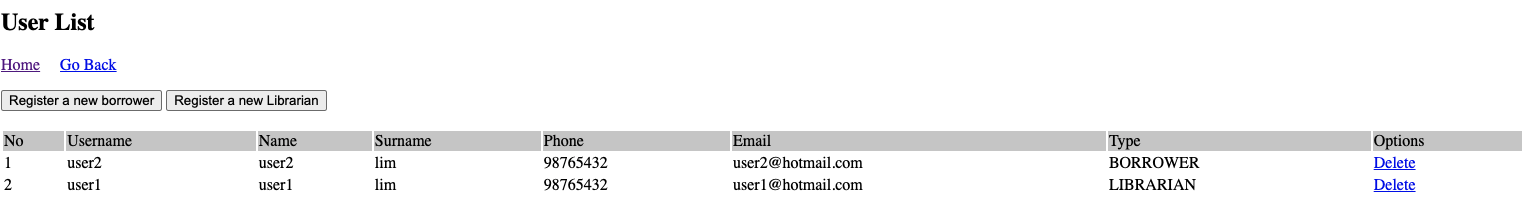
The edit.php allows the librarian to update existing resources that have been added into the database. After the record have been added, you can only make changes to status, rcost(regular cost) and ecost(extended cost). The same logic applies for rcost cannot be greater than ecost. After a successful update of the records, then the application will automatically redirect back to the previous page, with the records updated.



Resourcelist.php – resource updated successfully page

**5. /usermgt/userlist.php**

The user list page allows the librarian to add new librarian or new borrower. Sometimes, the librarian needs to help the user to add on behalf of new borrower perhaps the user is old and might not know how to own self enter the details or may need help. Alternatively, librarians can help to add new librarians into the systems.



Userlist.php – view user list

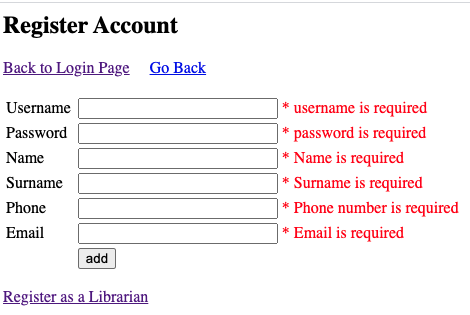
Without any login, both borrowers and user can create new accounts with from the index.html page by clicking on the registration button.

**6. /usermgt/ Register\_borrower.php**

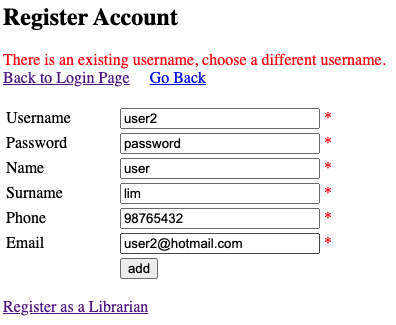
This is the page used for the creation of new borrower accounts. Username and password are added so that it is easy to manage user login. Validation have been added to this page so that

It prevent insertion of invalid data.

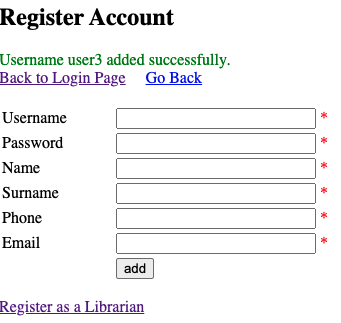
If the new librarian would like to register for new account can click on the register as a librarian link.



Register\_borrower.php – Error Validation



Register\_borrower.php – Prevent adding records of existing username

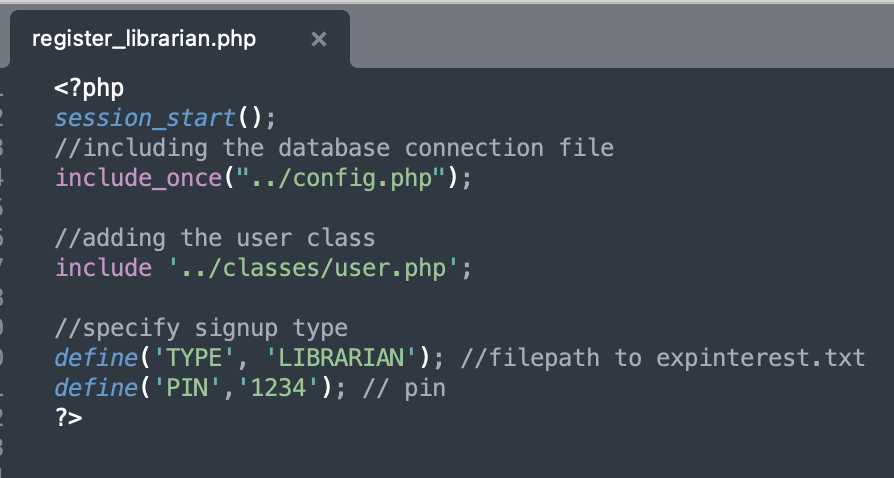


Register\_borrower.php – Username added successfully

**7. /usermgt/ Register\_librarian.php**

User will need to enter the pin in order to be able to have access to create librarian account. Currently, this PIN is set on the top of the register\_librarian.php page. This is to simulate in the event of the real life account setup for librarian administrator that they can still create a new account without the need of another librarian with the use of PIN.

In an actual situation, the pin should be stored in the database so that it be easy to manage configuration like this through a configuration table.

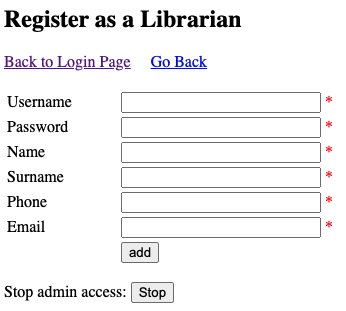


Register\_librarian.php – PIN set using the define function

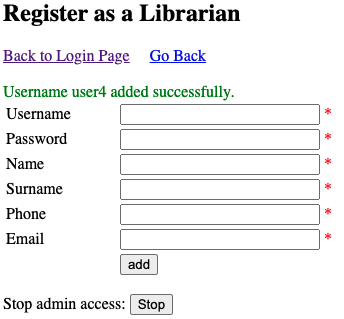


Register\_librarian.php – Logic to display / hide the UI on the page

User is able to add new librarian now from this page. The new librarian must remember to click on the stop button to stop the access to this page. Otherwise, another user that is not a librarian which does not have access to this page and does not know the pin will be able to create librarian access for themselves.



register\_librarian.php – User can click stop to close access

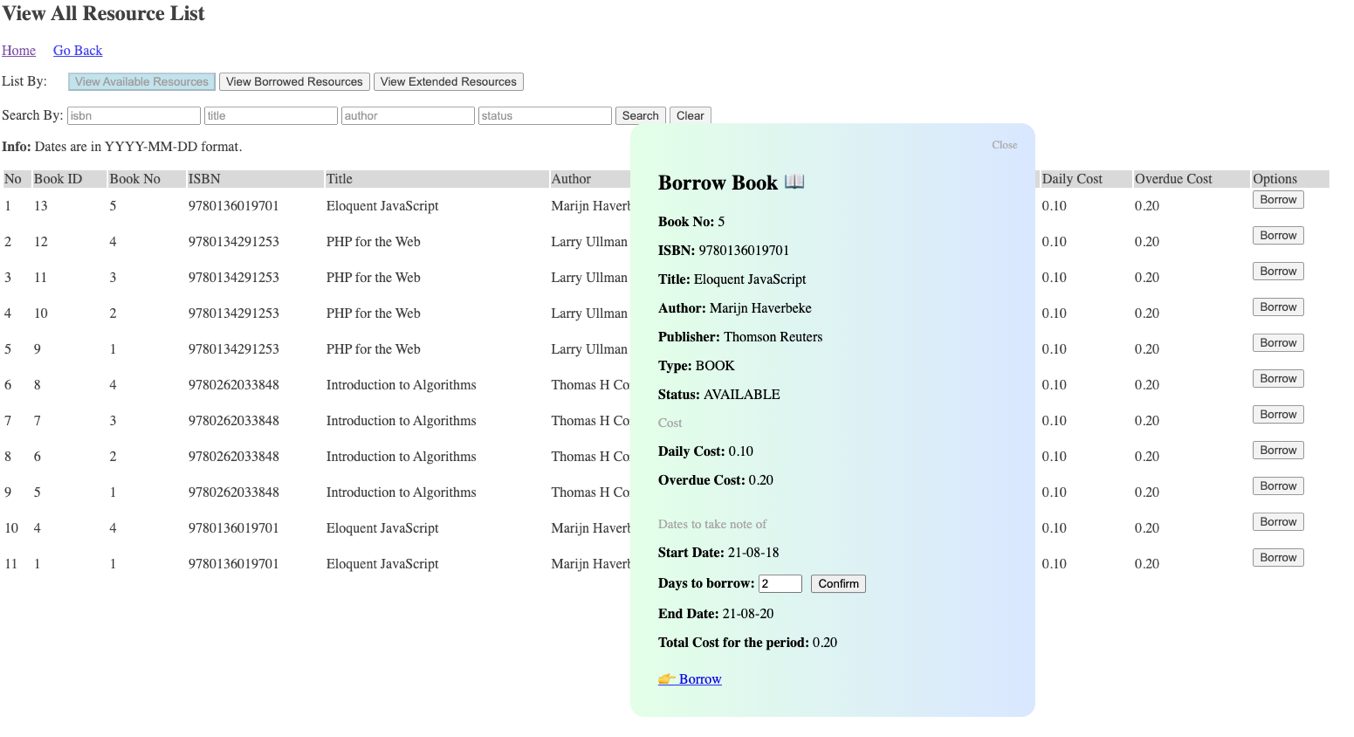


register\_librarian.php – Successful username added

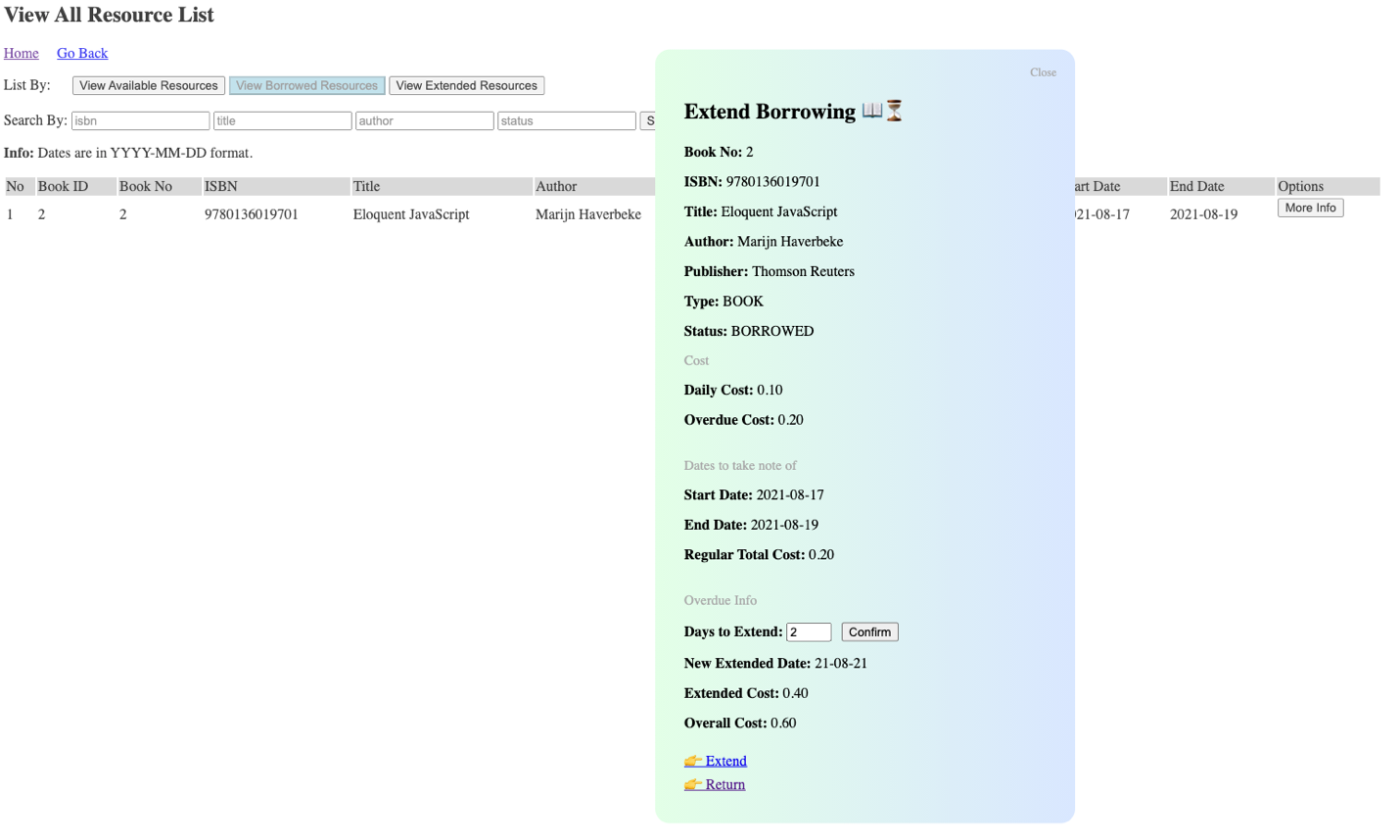


register\_librarian.php – After clicking stop, return to enter pin section

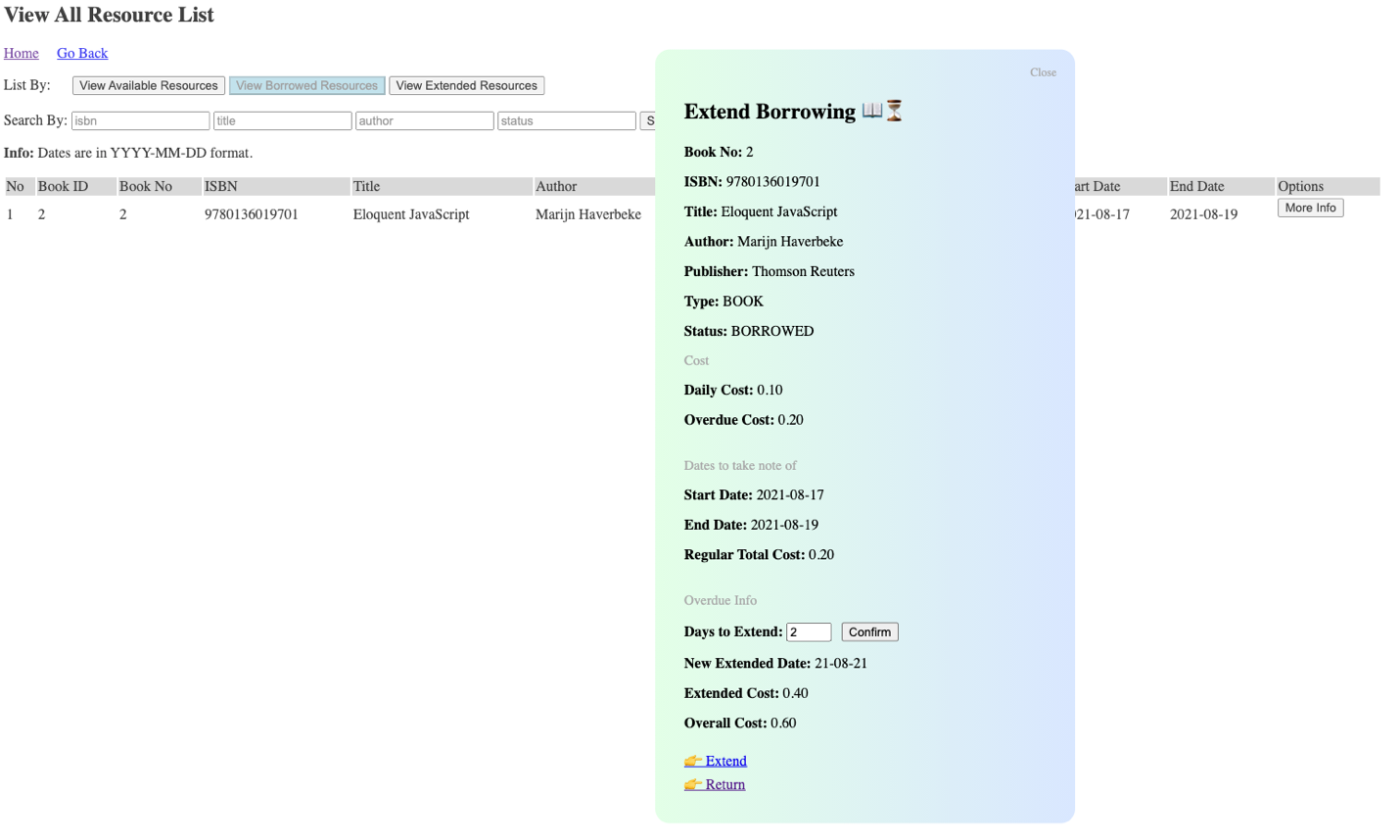
**8. resourcemgt/borrower/ availableresourcelist.php**This is the page used by borrowers. Borrowers can view available / borrowed / extended resources all in the same page. The logic for the searching is same as the librarian resourcelist.php page.



availableresourcelist.php – borrow notification after entering the days to borrow



availableresourcelist.php – extend borrowing notification after entering the days to borrow



availableresourcelist.php – view extended notification