
Student 1

Name: Sayyid Iskandar Khan

Student number: 6959751

Email address: sisoa078@uowmail.edu.au

Assignment number: 3

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) usermgmt/userlist.php – shows all the user accounts (only accessible via librarian)
- 4) usermgmt/register_borrower.php – signup as a borrower
- 5) usermgmt/register_librarian.php – signup as a librarian
- 6) usermgmt/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_resources.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 resources 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

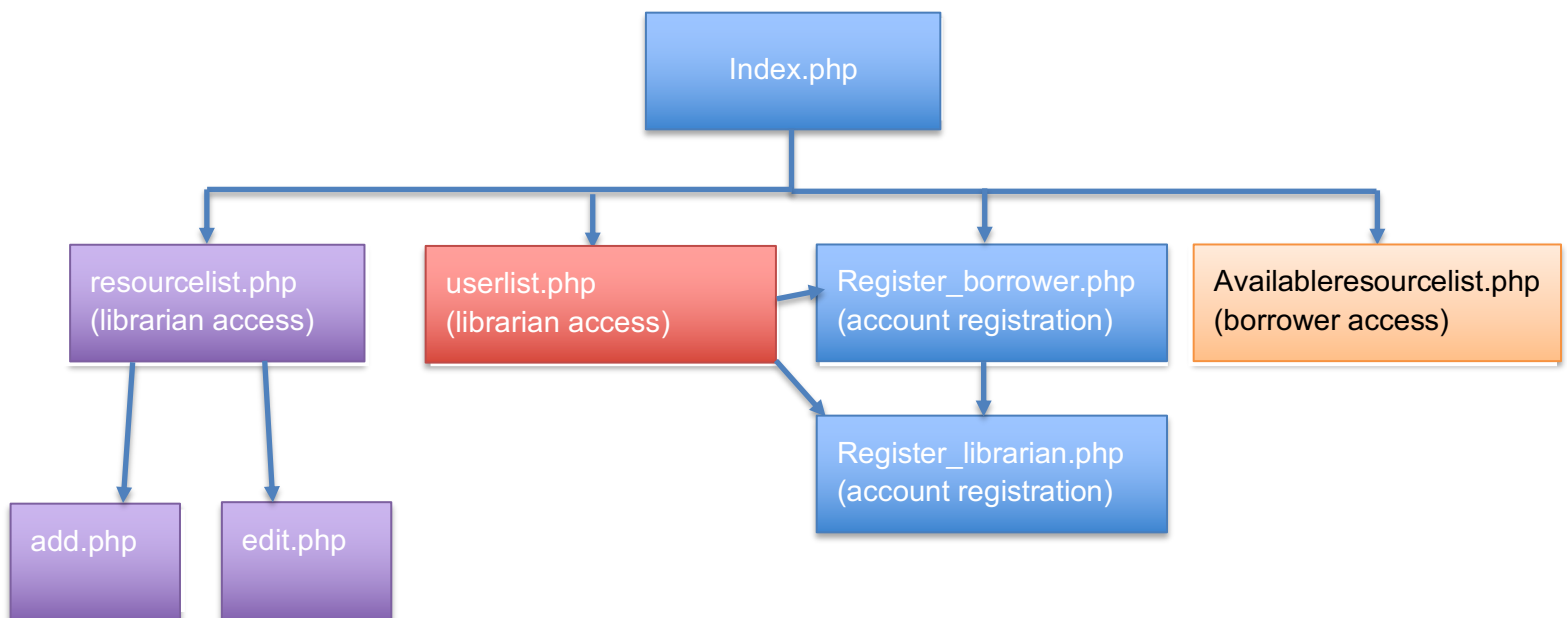
25) css/tableui.css - css styling for tables used in the pages

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.



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.

```
<?php
/**
 *change configuration information here only
 */
$databaseHost = 'localhost';
$databaseName = 'takoko_library';
$databaseUsername = 'root';
$databasePassword = 'root';
/**
 *change configuration information here only
 */

// 1. Connect to MySQL
$mysqli = new mysqli($databaseHost, $databaseUsername, $databasePassword);
if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}

// 2. If database is not exist create one
if (!$mysqli_select_db($mysqli,$databaseName)){
    $sql = "CREATE DATABASE ".$databaseName;
    if ($mysqli->query($sql) === TRUE) {
        echo "Database initialised successfully";
    }
    else {
        echo "Error creating database: " . $mysqli->error;
    }
}

// 3. create tables if they don't exist
function initTableIfNotExist($conn,$table_name,$sql_filename) {
    if ( $conn->query("select 1 from $table_name limit 1;") ) {
        // my_table exists, no action required'
    }
    else {
        //table not exist, create tables
        $query = file_get_contents($sql_filename);
        $conn->multi_query($query);
    }
}

/**
 *load all table name below
 */
//parameters to pass: ($conn,$table_name,$sql_filename)
initTableIfNotExist($mysqli, table_name: "users", sql_filename: "database/users.sql");
initTableIfNotExist($mysqli, table_name: "resources", sql_filename: "database/resources.sql");
?>
```

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_resources.php

-> database/dataset_users.php

To pre-load dataset to expedite testing. Navigate to 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).

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment  
  `username` varchar(50) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `surname` varchar(50) NOT NULL,  
  `phone` varchar(50) NOT NULL,  
  `email` varchar(100) NOT NULL,  
  `type` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

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.

```
CREATE TABLE `resources` (  
  `bookid` int(11) NOT NULL auto_increment,  
  `bookno` int(2) NOT NULL,  
  `isbn` varchar(50) NOT NULL,  
  `title` varchar(50) NOT NULL,  
  `author` varchar(20) NOT NULL,  
  `publisher` varchar(50) NOT NULL,  
  `type` varchar(50) NOT NULL,  
  `status` varchar(50) NOT NULL,  
  `rcost` DECIMAL(5,2) NOT NULL,  
  `ecost` DECIMAL(5,2) NOT NULL,  
  `username` varchar(50),  
  `startdate` DATE,  
  `enddate` DATE,  
  `extenddate` DATE,  
  PRIMARY KEY (`bookid`)  
);
```

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.

View All Resource List

[Home](#) [Go Back](#)

you have no access to this section of the page.

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

Takoko Digital Library

Welcome to Takoko Digital Library. Enter your login details to proceed

Hello LIBRARIAN!

Select the options below:

Menu access

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

Takoko Digital Library

Welcome to Takoko Digital Library. Enter your login details to proceed

Hello User!

Select the options below:

View / 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.

```
<?php
//for detecting empty value (search button clicked but query is empty)
if (isset($_POST['library-login']) && (empty($_POST['login']) || empty($_POST['password']))) {
    $_SESSION['login'] = '';
    session_unset();
    if(empty($_POST['login'])){
        echo '*username cannot be blank';
    }
    else if(empty($_POST['password'])){
        echo '*password cannot be blank';
    }
}
elseif(isset($_POST['library-login']) && !empty($_POST['login'])) {
    $validateLogin = $_POST['login'];

    //////////// validate login using SQL ////////////
    $query = mysqli_query($mysqli, query: "SELECT * FROM users WHERE username = '$validateLogin'");
    $checkExistingUsername = mysqli_fetch_array($query);
    //if not empty - proceed with further validation
    if(!empty($checkExistingUsername)) {
        $userObj = User::init($checkExistingUsername);
        $pw = $userObj->password;
        $type = $userObj->type;
        if($_POST['password'] == $pw) {
            $_SESSION['login'] = $validateLogin;
            $_SESSION['type'] = $type;
            //refresh UI to update session
            header( string: "Refresh: 0.1");
        }
        else {
            $_SESSION['login'] = '';
            $_SESSION['type'] = '';
            session_unset();
            echo '*incorrect password ';
        }
    }
    else {
        $_SESSION['login'] = '';
        $_SESSION['type'] = '';
        session_unset();
        echo '*username does not exist or not in system';
    }
    //////////// validate login using SQL ////////////
}
elseif(isset($_POST['library-logout'])) {
    unset($_SESSION['login']);
    unset($_SESSION['type']);
    session_destroy();
    header( string: "Refresh: 0.1");
}
?>
```

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.

Takoko Digital Library

Welcome to Takoko Digital Library. Enter your login details to proceed

Dont Have an account? [Register](#) with us today!

© 2021 All rights reserved.

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.

Takoko Digital Library

Welcome to Takoko Digital Library. Enter your login details to proceed

Hello User!

Select the options below:

© 2021 All rights reserved.

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.

View All Resource List

[Home](#) [Go Back](#)

Page Navigation: [Add a new resource](#)

List By: [View All Resources](#) [View Available Resources](#) [View Borrowed Resources](#) [View Extended Resources](#)

Search By: [Search](#) [Clear](#)

No	Book ID	Book No	ISBN	Title	Author	Publisher	Type	Status	R - Cost	E - Cost	Options
1	12	4	9780134291253	PHP for the Web	Larry Ullman	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
2	11	3	9780134291253	PHP for the Web	Larry Ullman	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
3	10	2	9780134291253	PHP for the Web	Larry Ullman	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
4	9	1	9780134291253	PHP for the Web	Larry Ullman	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
5	8	4	9780262033848	Introduction to Algorithms	Thomas H Cormen	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
6	7	3	9780262033848	Introduction to Algorithms	Thomas H Cormen	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
7	6	2	9780262033848	Introduction to Algorithms	Thomas H Cormen	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
8	5	1	9780262033848	Introduction to Algorithms	Thomas H Cormen	publisher	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
9	4	4	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete
10	3	3	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	EXTENDED	0.10	0.20	More info Update Status Delete
11	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	BORROWED	0.10	0.20	More info Update Status Delete
12	1	1	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More info Update Status Delete

Overview of resourcelist.php

View All Resource List

[Home](#) [Go Back](#)

Page Navigation: [Add a new resource](#)

List By: [View All Resources](#) [View Available Resources](#) [View Borrowed Resources](#) [View Extended Resources](#)

Search By: [S](#)

No	Book ID	Book No	ISBN	Title	Author
1	12	4	9780134291253	PHP for the Web	Larry Ullman
2	11	3	9780134291253	PHP for the Web	Larry Ullman
3	10	2	9780134291253	PHP for the Web	Larry Ullman
4	9	1	9780134291253	PHP for the Web	Larry Ullman
5	8	4	9780262033848	Introduction to Algorithms	Thomas H Cormen
6	7	3	9780262033848	Introduction to Algorithms	Thomas H Cormen
7	6	2	9780262033848	Introduction to Algorithms	Thomas H Cormen
8	5	1	9780262033848	Introduction to Algorithms	Thomas H Cormen
9	4	4	9780136019701	Eloquent JavaScript	Marijn Haverbeke
10	3	3	9780136019701	Eloquent JavaScript	Marijn Haverbeke
11	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke
12	1	1	9780136019701	Eloquent JavaScript	Marijn Haverbeke

Book Info

Book No: 4

ISBN: 9780134291253

Title: PHP for the Web

Author: Larry Ullman

Publisher: publisher

Type: BOOK

Status: AVAILABLE

Cost

Daily Cost: 0.10

Overdue Cost: 0.20

Dates to take note of

Start Date: 2021-08-18

End Date: 2021-08-20

Regular Total Cost: 0.20

Overdue Info

Extended Date: 2021-08-22

Extended Cost: 0.40

Overall Cost: 0.60

Cost	Options
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete
0	More info Update Status Delete

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.

```
function allNotification($all_modal_id,$resource) {
    function diff_in_days($e,$l) {
        $earlier = new DateTime($e);
        $later = new DateTime($l);

        $abs_diff = $later->diff($earlier)->format("%a");
        return $abs_diff;
    }

    function isEmpty($value) {
        return empty($value) ? "N/A" : $value;
    }

    function isZero($value) {
        return floatval($value) > 0 ? number_format($value,2) : '0';
    }

    $bookID = $resource->bookid;
    $total_regular_cost = isZero(number_format(diff_in_days($resource->startdate,$resource->enddate) * $resource->rcost, 2));

    $diff_between_enddate_and_extenddate = diff_in_days($resource->enddate,$resource->extenddate);
    $extend_cost = isZero(number_format(intval($diff_between_enddate_and_extenddate) * $resource->ecost, 2));
    $total_cost = isZero(number_format(floatval($extend_cost) + floatval($total_regular_cost), 2));

    $startdate = isEmpty($resource->startdate);
    $enddate = isEmpty($resource->enddate);
    $extenddate = isEmpty($resource->extenddate);

    $message =
    "
        <div id='$all_modal_id' class='modal-window'>
            <div>
                <a href='#' title='Close' class='modal-close'>Close</a>
                <h2>Book Info ■■</h2>
                <div><b>Book No: </b> $resource->bookno</div>
                <div><b>ISBN: </b> $resource->isbn</div>
                <div><b>Title: </b> $resource->title</div>
                <div><b>Author: </b> $resource->author</div>
                <div><b>Publisher: </b> $resource->publisher</div>
                <div><b>Type: </b> $resource->type</div>
                <div><b>Status: </b> $resource->status</div>

                <div><small style='color: darkgrey;font-size: 15px;'>Cost </small></div>
                <div><b>Daily Cost: </b> $resource->rcost</div>
                <div><b>Overdue Cost: </b> $resource->ecost</div>
                <br>

                <div><small style='color: darkgrey;font-size: 15px;'>Dates to take note of </small></div>
                <div><b>Start Date: </b> $startdate</div>
                <div><b>End Date: </b> $enddate</div>
                <div><b>Regular Total Cost: </b> $total_regular_cost</div>
                <br>

                <div><small style='color: darkgrey;font-size: 15px;'>Overdue Info </small></div>
                <div><b>Extended Date: </b> $extenddate</div>
                <div><b>Extended Cost: </b> $extend_cost</div>
                <div><b>Overall Cost: </b> $total_cost</div>
                <br>

                <br>
            </div>
        </div>
    ";
    return $message;
}
```

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.

```
$result = array();
$query = '';

function getCurrentSection($type_of_query) {
    $current_section = $GLOBALS['current_section'];
    $additional_query = '';
    //for multiple filters
    if($type_of_query === 'MULTIPLE_FILTER') {
        $additional_query = ' AND status = ';
    }
    //for single filter only
    else if($type_of_query === 'SINGLE_FILTER') {
        $additional_query = ' WHERE status = ';
    }
}

if($current_section === 'ALL') {
    $additional_query = '';
}
else if($current_section === 'AVAILABLE') {
    $additional_query .= "'AVAILABLE'";
}
else if($current_section === 'BORROWED') {
    $additional_query .= "'BORROWED'";
}
else if($current_section === 'EXTENDED') {
    $additional_query .= "'EXTENDED'";
}
return $additional_query;
}
```

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.

```
// logic for search query
if(!empty($_GET['isbn']) || !empty($_GET['title']) || !empty($_GET['author']) || !empty($_GET['status'])) {
    $sql_statement = "SELECT * FROM resources WHERE ";

    //1. put all the getters in the array
    $filter_array = array(
        'isbn' => $_GET['isbn'],
        'title' => $_GET['title'],
        'author' => $_GET['author'],
        'status' => $_GET['status'],
    );
    //2. init a query array
    $query_array = array();
    //3. loop through the array, if value is not null, add it to the query array
    foreach ($filter_array as $key => $value) {
        if ($value != '') {
            //match using like clause
            $query_array[] = $key . ' LIKE ' . "'" . "%" . $value . "%" . "'";
        }
    }
    //4. join all the statement with the AND statement
    $sql_statement = "SELECT * FROM resources WHERE " . implode( ' AND ', $query_array);
    //5. depending on the current_section, it will filter based on the availability of the books
    $sql_statement .= getCurrentSection( type_of_query: 'MULTIPLE_FILTER');
    //6. add sort by desc
    $sql_statement .= " ORDER BY bookid DESC";
    //7. generate the query
    $query = mysqli_query($mysqli,$sql_statement);
}
else {
    //1. fetching data in descending order (latest entry first)
    $sql_statement = "SELECT * FROM resources ";
    //2. depending on the current_section, it will filter based on the availability of the books
    $sql_statement .= getCurrentSection( type_of_query: 'SINGLE_FILTER');
    //3. add sort by desc
    $sql_statement .= " ORDER BY bookid DESC";
    $query = mysqli_query($mysqli, $sql_statement);
}
```

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.

```
while($res = mysqli_fetch_array($query)){
    //while there is still data to query, continue query until no more result
    array_push( &array: $result, Resources::init($res));
}
```

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.

← → ↺ ① localhost:8888/php_library/resourcemgmt/librarian/resourcelist.php?isbn=&title=Eloquent+JavaScript&author=Marijn&status=¤t_section=ALL#

View All Resource List

[Home](#) [Go Back](#)

Page Navigation:

List By:

Search By:

No	Book ID	Book No	ISBN	Title	Author	Publisher	Type	Status	R - Cost	E - Cost	Options
1	13	5	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete
2	4	4	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete
3	3	3	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	EXTENDED	0.10	0.20	More Info Update Status Delete
4	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	BORROWED	0.10	0.20	More Info Update Status Delete
5	1	1	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete

Resourcelist.php - View All Resources

View All Resource List

[Home](#) [Go Back](#)

Page Navigation:

List By:

Search By:

No	Book ID	Book No	ISBN	Title	Author	Publisher	Type	Status	R - Cost	E - Cost	Options
1	13	5	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete
2	4	4	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete
3	1	1	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	AVAILABLE	0.10	0.20	More Info Update Status Delete

Resourcelist.php - View Available Resources

View All Resource List

[Home](#) [Go Back](#)

Page Navigation:

List By:

Search By:

No	Book ID	Book No	ISBN	Title	Author	Publisher	Type	Status	R - Cost	E - Cost	Options
1	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	BORROWED	0.10	0.20	More Info Update Status Delete

Resourcelist.php - View Borrowed Resources

View All Resource List

[Home](#) [Go Back](#)

Page Navigation:

List By:

Search By:

No	Book ID	Book No	ISBN	Title	Author	Publisher	Type	Status	R - Cost	E - Cost	Options
1	3	3	9780136019701	Eloquent JavaScript	Marijn Haverbeke	Thomson Reuters	BOOK	EXTENDED	0.10	0.20	More Info Update Status Delete

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.

```
<?php
//including the database connection file
include("../..//config.php");

//getting id of the data from url
$bookid = $_GET['bookid'];

//deleting the row from table
$result = mysqli_query($mysqli, "DELETE FROM resources WHERE bookid=$bookid");

//redirecting to the display page (userlist.php in our case)
header("Location: resourcelist.php");
?>
```

[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 a new resource

[Back to Resource list](#) [Go Back](#)

Book No	<input type="text"/>	*
ISBN	<input type="text"/>	*
Title	<input type="text"/>	*
Author	<input type="text"/>	*
Publisher	<input type="text"/>	*
Type	<input type="text" value="v"/>	*
Regular Cost	<input type="text"/>	*
Extended Cost	<input type="text"/>	*
<input type="button" value="add"/>		

INFO:

- ISBN number is stored in digits only without dashes. You may put dashes, app will store digits only.
- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as Extended cost.

Insert new resource page when visited

Insert a new resource

There is an existing isbn with the same book no, increment either the book no or set a different isbn.

[Back to Resource list](#) [Go Back](#)

Book No	<input type="text" value="1"/>	*
ISBN	<input type="text" value="9780136019701"/>	*
Title	<input type="text" value="Eloquent JavaScript"/>	*
Author	<input type="text" value="Marijn Haverbeke"/>	*
Publisher	<input type="text" value="Thomson Reuters"/>	*
Type	<input type="text" value="BOOK"/>	*
Regular Cost	<input type="text" value="0.10"/>	*
Extended Cost	<input type="text" value="0.20"/>	*
<input type="button" value="add"/>		

INFO:

- ISBN number is stored in digits only without dashes. You may put dashes, app will store digits only.
- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as Extended cost.

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

Insert a new resource

[Back to Resource list](#) [Go Back](#)

Book No	<input type="text"/>	* Book No is required
ISBN	<input type="text"/>	* ISBN is required
Title	<input type="text"/>	* Title is required
Author	<input type="text"/>	* Author is required
Publisher	<input type="text"/>	* Publisher is required
Type	<input type="text" value="v"/>	* Type is required
Regular Cost	<input type="text"/>	* Cost is required
Extended Cost	<input type="text"/>	* Cost is required
<input type="button" value="add"/>		

INFO:

- ISBN number is stored in digits only without dashes. You may put dashes, app will store digits only.
- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as Extended cost.

Validation added to prevent incorrect data inserted

[Back to Resource list](#) [Go Back](#)

Book No	<input type="text" value="1"/>	*
ISBN	<input type="text" value="9780136019701"/>	*
Title	<input type="text" value="Eloquent JavaScript"/>	*
Author	<input type="text" value="Marijn Haverbeke"/>	*
Publisher	<input type="text" value="Thomson Reuters"/>	*
Type	<input type="text" value="BOOK"/>	*
Regular Cost	<input type="text" value="0.20"/>	*
Extended Cost	<input type="text" value="0.10"/>	* cost set too low
<input type="button" value="add"/>		

INFO:

- ISBN number is stored in digits only without dashes. You may put dashes, app will store digits only.
- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as Extended cost.

Validation to prevent regular cost greater than extended cost

Insert a new resource

Resource added successfully.

[Back to Resource list](#) [Go Back](#)

Book No	<input type="text"/>	*
ISBN	<input type="text"/>	*
Title	<input type="text"/>	*
Author	<input type="text"/>	*
Publisher	<input type="text"/>	*
Type	<input type="text" value="BOOK"/>	*
Regular Cost	<input type="text"/>	*
Extended Cost	<input type="text"/>	*
<input type="button" value="add"/>		

INFO:

- ISBN number is stored in digits only without dashes. You may put dashes, app will store digits only.
- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as 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.

Resource updated successfully.

[Back to Resource List](#) [Go Back](#)

Book ID	<input type="text" value="12"/>
Book No	<input type="text" value="4"/>
ISBN	<input type="text" value="9780134291253"/>
Title	<input type="text" value="PHP for the Web"/>
Author	<input type="text" value="Larry Ullman"/>
Publisher	<input type="text" value="publisher"/>
Type	<input type="text" value="BOOK"/>
Status	<input type="text" value="AVAILABLE"/> *
Regular Cost	<input type="text" value="0.10"/> *
Extended Cost	<input type="text" value="0.20"/> *
<input type="button" value="Update"/>	

INFO:

- Max Amount for Regular Cost or Extended Cost is 1.00 .
- Extended Cost have to be set higher than Regular cost . Regular cost can be the same as Extended cost.

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.

User List

[Home](#) [Go Back](#)

[Register a new borrower](#) [Register a new Librarian](#)

No	Username	Name	Surname	Phone	Email	Type	Options
1	user2	user2	lim	98765432	user2@hotmail.com	BORROWER	Delete
2	user1	user1	lim	98765432	user1@hotmail.com	LIBRARIAN	Delete

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 Account

[Back to Login Page](#) [Go Back](#)

Username	<input type="text"/>	* username is required
Password	<input type="text"/>	* password is required
Name	<input type="text"/>	* Name is required
Surname	<input type="text"/>	* Surname is required
Phone	<input type="text"/>	* Phone number is required
Email	<input type="text"/>	* Email is required
<input type="button" value="add"/>		

[Register as a Librarian](#)

Register_borrower.php – Error Validation

Register Account

There is an existing username, choose a different username.

[Back to Login Page](#) [Go Back](#)

Username	<input type="text" value="user2"/>	*
Password	<input type="text" value="password"/>	*
Name	<input type="text" value="user"/>	*
Surname	<input type="text" value="lim"/>	*
Phone	<input type="text" value="98765432"/>	*
Email	<input type="text" value="user2@hotmail.com"/>	*
<input type="button" value="add"/>		

[Register as a Librarian](#)

Register_borrower.php – Prevent adding records of existing username

Register Account

Username user3 added successfully.

[Back to Login Page](#) [Go Back](#)

Username	<input type="text"/>	*
Password	<input type="text"/>	*
Name	<input type="text"/>	*
Surname	<input type="text"/>	*
Phone	<input type="text"/>	*
Email	<input type="text"/>	*
<input type="button" value="add"/>		

[Register as a Librarian](#)

Register_borrower.php – Username added successfully

7. /usermgmt/ 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 x
<?php
session_start();
//including the database connection file
include_once("../config.php");

//adding the user class
include '../classes/user.php';

//specify signup type
define('TYPE', 'LIBRARIAN'); //filepath to expinterest.txt
define('PIN', '1234'); // pin
?>
```

Register_librarian.php – PIN set using the define function

```
<?php
if(isset($_POST['verify']) && $_POST['verify'] == 'verify') {
    $currentPin = PIN;
    if( isset($_POST['admin_pin'])) {
        if($_POST['admin_pin'] == $currentPin) {
            $_SESSION['admin_pin'] = 'success';
        }
        else {
            $_SESSION['admin_pin'] = '';
            unset($_SESSION['admin_pin']);
        }
    }
    else {
        $_SESSION['admin_pin'] = '';
        unset($_SESSION['admin_pin']);
    }
}
```

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 as a Librarian

[Back to Login Page](#) [Go Back](#)

Username	<input type="text"/>	*
Password	<input type="password"/>	*
Name	<input type="text"/>	*
Surname	<input type="text"/>	*
Phone	<input type="text"/>	*
Email	<input type="text"/>	*
<input type="button" value="add"/>		

Stop admin access:

register_librarian.php – User can click stop to close access

Register as a Librarian

[Back to Login Page](#) [Go Back](#)

Username user4 added successfully.

Username	<input type="text"/>	*
Password	<input type="password"/>	*
Name	<input type="text"/>	*
Surname	<input type="text"/>	*
Phone	<input type="text"/>	*
Email	<input type="text"/>	*
<input type="button" value="add"/>		

Stop admin access:

register_librarian.php – Successful username added

Register as a Librarian

[Back to Login Page](#) [Go Back](#)

Enter Pin:

register_librarian.php – After clicking stop, return to enter pin section

8. resourcemgt/borrower/ availablesourcelist.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.

View All Resource List

[Home](#) [Go Back](#)

List By: [View Available Resources](#) [View Borrowed Resources](#) [View Extended Resources](#)

Search By:

Info: Dates are in YYYY-MM-DD format.

No	Book ID	Book No	ISBN	Title	Author
1	13	5	9780136019701	Eloquent JavaScript	Marijn Haverbeke
2	12	4	9780134291253	PHP for the Web	Larry Ullman
3	11	3	9780134291253	PHP for the Web	Larry Ullman
4	10	2	9780134291253	PHP for the Web	Larry Ullman
5	9	1	9780134291253	PHP for the Web	Larry Ullman
6	8	4	9780262033848	Introduction to Algorithms	Thomas H Cormen
7	7	3	9780262033848	Introduction to Algorithms	Thomas H Cormen
8	6	2	9780262033848	Introduction to Algorithms	Thomas H Cormen
9	5	1	9780262033848	Introduction to Algorithms	Thomas H Cormen
10	4	4	9780136019701	Eloquent JavaScript	Marijn Haverbeke
11	1	1	9780136019701	Eloquent JavaScript	Marijn Haverbeke

Borrow Book

Book No: 5
ISBN: 9780136019701
Title: Eloquent JavaScript
Author: Marijn Haverbeke
Publisher: Thomson Reuters
Type: BOOK
Status: AVAILABLE
Cost:
Daily Cost: 0.10
Overdue Cost: 0.20
Dates to take note of
Start Date: 21-08-18
Days to borrow:
End Date: 21-08-20
Total Cost for the period: 0.20
[📖 Borrow](#)

Daily Cost	Overdue Cost	Options
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>
0.10	0.20	<input type="button" value="Borrow"/>

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

View All Resource List

[Home](#) [Go Back](#)

List By: [View Available Resources](#) [View Borrowed Resources](#) [View Extended Resources](#)

Search By:

Info: Dates are in YYYY-MM-DD format.

No	Book ID	Book No	ISBN	Title	Author
1	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke

Extend Borrowing

Book No: 2
ISBN: 9780136019701
Title: Eloquent JavaScript
Author: Marijn Haverbeke
Publisher: Thomson Reuters
Type: BOOK
Status: BORROWED
Cost:
Daily Cost: 0.10
Overdue Cost: 0.20
Dates to take note of
Start Date: 2021-08-17
End Date: 2021-08-19
Regular Total Cost: 0.20
Overdue Info
Days to Extend:
New Extended Date: 21-08-21
Extended Cost: 0.40
Overall Cost: 0.60
[📖 Extend](#)
[📖 Return](#)

Start Date	End Date	Options
2021-08-17	2021-08-19	<input type="button" value="More Info"/>

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

View All Resource List

[Home](#) [Go Back](#)

List By: [View Available Resources](#) [View Borrowed Resources](#) [View Extended Resources](#)

Search By:

Info: Dates are in YYYY-MM-DD format.

No	Book ID	Book No	ISBN	Title	Author
1	2	2	9780136019701	Eloquent JavaScript	Marijn Haverbeke

Extend Borrowing

Close

Book No: 2

ISBN: 9780136019701

Title: Eloquent JavaScript

Author: Marijn Haverbeke

Publisher: Thomson Reuters

Type: BOOK

Status: BORROWED

Cost

Daily Cost: 0.10

Overdue Cost: 0.20

Dates to take note of

Start Date: 2021-08-17

End Date: 2021-08-19

Regular Total Cost: 0.20

Overdue Info

Days to Extend:

New Extended Date: 21-08-21

Extended Cost: 0.40

Overall Cost: 0.60

 [Extend](#)

 [Return](#)

Start Date	End Date	Options
2021-08-17	2021-08-19	<input type="button" value="More Info"/>

availablesourcelist.php – view extended notification