Library website with db access by jakub zareba

List of files/pages used:

Connection.php

* This page contains simple script that connects web page to a database and can perform operations (queries) on said database. Every page has to contain **require\_once "connection.php";** function in order to gain access to a database without recopying entire script into each single page.

Site.css

* This page contains all the visual enchantments of every visible element on the site, rendering it more visually appealing to the user.

Index.html

* This page just contains two links for users, either to go and register or log in to the home page.

Registration.html

* This page displays to the user a simple form that offers user to create an account in the library system. It contains few input fields and submit button that sends data entered by the user to appropriate php scripts that perform queries on the data and returns back results, on which depends if user account creation went smoothly. If not, data won’t be added to the data base and appropriate error will be displayed to the user.

Registration.php

* This is the page that receives data from registration.html. It contains scripts that run queries on received data and performs all the validation needed in order to prevent users from not entering password, failing password confirmation or creating username that already exists in the database.

Print\_errors.php

* It’s a page that receives data from few other pages and its sole purpose is to print error messages it receives. Also prints success messages after user performs an action that ends in a success, such as successful book reservation, login, account creation or removing reservation.

Login.html

* The page is similar to registration.html. All it offers is few input fields that will carry user data to php scripts and a submit button. It also contains a return link back to the index page.

Login.php

* This script applies few error checking measures on the data received from user in order to ensure it was entered properly (i.e. no empty password) or if the username and password match within the database. If data entered matches the database, it will allow user into the home page, if not it will refresh the page and print appropriate errors.

Home.php

* Home.php is a mixture of php and html in order to present an average looking interface to the user. It contains radio buttons that allow the user to select appropriate searching method and enter a book title or author’s name. Also, a drop down menu with category selection that will display all the books from given category to the user. This is the first page on which header displays the username that was assigned during the logging in process to a session variable. Also within the header, under the username, a logout button and reservation button are present. First one, logs out the user and destroys the session so the variables can be reused in a later session by the next user. After data is entered by the user it’s being carried to appropriate scripts.

User\_search.php

* This is a page that contains error checking measures that ensure which radio button was selected so appropriate query can be run. After query is done, retrieved database data acquired from the query is assigned to a global variable in form of array. Then sends the data to a page designed for printing out arrays like the one we just created.

Display\_search\_results.php

* This is the page that will receive query data from every script on the website in order to print it nicely in a table. As book records are stored in a form of 2D arrays, the script contains a nested for loop that prints out table which contains the data that was sent by another script. As we were asked to display data only 5 elements per page, it also contains a simple algorithm that calculates the variables which are used as a loop boundaries specifiers
* There are 3 buttons placed under the freshly printed table. Each button contains a link that will take user to appropriate place on the website and will perform desired by the user action. First is the return link, which takes user back to the home page. Next two are previous and next page links. They have implemented error checking, so that if you are viewing the first 5 elements and you don’t have any previous elements yet, the button will not be displayed. Same with next page button.

Next\_page.php

* This script displays the next 5 elements of the search to the user. It again contains simple algorithm that keeps track of which 5 elements should be printed. It again prints table and has 3 links at the bottom of the table that allow user to undertake desired action.

Previous\_page.php

* This script pretty much reverses everything that next\_page.php did that so the script can print 5 previous results of the search to the user. It as well contains variables tracking correct position within the list so it can print correct 5 elements and keep correct indexing.

Category\_list.php

* This page is pretty much the same thing as user\_search.php. The only difference is that user wanted to display books by the category, so the script runs different query and then in a same way assigns the query result into an array, which is then assigned to a global variable and carried out to display\_search\_results.php in order to be printed.

Logout.php

* This script is loaded after user clicks the logout button present on every page after successful log in, placed in the header below the username. All it does is unsetting the session variables so they don’t store any information from previous session after another user logs in.

Display\_reservations.php

* In this script we have only few lines of code that does pretty much what user\_search.php and display\_search\_results do. It checks the username of the user and runs a query to check all the books resvered by the user within the database.

Reserve.php

* This script is a collection of queries, starting with retrieving book’s ISBN associated with the reserve button that was clicked by the user and started this script. Then simple query is run to grab full row of data that belongs to this specific book ISBN. Then script checks the number of how many books the user has already reserved and updates it by one. The last query inserts a new record into the reservation table within the database. Also, it updates the books table within the database to change the value to “Yes” as in reserved. The new insert contains the previously retrieved book’s ISBN, username of the user reserving the book and date in format dd/mm/yyyy. Then the appropriate message is being displayed to the user depending if the operation was successful and user is being redirected to the home page on which he/her can perform a new search.

Unreserve.php

* This script reverses all the changes made by reserve.php. It decreases the counter of user’s currently reserved books, it removes the book from user’s reserved list, it changes the position reserved in the book table back to “No”.