home.php

sanitizeInput(String $data){

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

bookSearch(String $searchTerm){

takes the inputted $searchTerm and compares it to the xml file and gives suggestions.

Returns $searchTerm as a string to fetchBooks() function.

}

fetchFeturedItems(Object $pdo, int $itemType){

prepares and executes SELECT query where item\_type\_fk = $itemType; to get the different featured items

output

returns results as array

}

populateFeaturedItemField(Object $feturedItems){

runs foreach loop for the previous query results

returns TRUE upon completion

}

books.php

fetchBooks(Object $pdo, String $searchTerm){

prepares and executes a very long SELECT query for books table that is inner joined with other tables such as book\_authors.

Returns results as an array

It sends back an array called $books upon completion.

}

populateBookField(Array $books){

runs a foreach loop for the previous query results

}

singleBook.php

fetchSingleBook(Object $pdo, Int $currBookId){

takes the books id and preforms a SELECT query to get all data on it

}

contact.php

sendContactRequest(Object $pdo){

takes the request data and does an INSERT query to put it in the database

Returns TRUE if successful, else it returns FALSE

}

employeeLogin.php

loginUser(Object $pdo){

preform a SELECT query based on the inputed username and saves data as $userData

if(Array !$userdata){

return "FALSE";

}

elseif(Array $userdata){

Verifyes that the password is correct

}

if(Bool $passwordMatch == TRUE){

saves relevant data like username, id and role number to their respective $\_SESSION variables

if(int $role = admin){

redirects to adminPanel.php

}

else{

redirect to bookEditor.php

}}}

adminPanel.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

bookEditor.php

//employee only functions

searchOwnBooks(Object $pdo, int $userId, $searchTerm){

preforms a SELECT query for books WHERE employee\_fk = $userId and WHERE book\_titel = $searchTerm

}

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

createNewBook(object $pdo){

preforms a INSERT query with the inputed data. Creates new rows in associated tables such as book\_author.

Returns true if successful

}

editBook(object $pdo, int $bookId){

preforms an UPDATE query with the new data WHERE book\_id = $bookId

returns true if successful

else returns false

}

deleteBook(object $pdo, int $bookId){

preforms a DELETE query on the books table WHERE book\_id = $bookId

returns TRUE when successful

}

sanitizeInput(string $data){

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

searchAllBooks(object $pdo, string $searchTerm){

preforms a SELECT query for books WHERE book\_titel = $searchTerm

returns books as an array

}

addAuthor(object $pdo, string $authorName, string $redirectLink){

preforms a INSERT query with the inputed data.

Stores success message and redirects upon completion

}

addIllustrator(object $pdo, string $illustratorName, string $redirectLink){

-||-

}

addPublisher(object $pdo, string $publisherName, string $redirectLink){

-||-

}

addSeries(object $pdo, string $seriesName, string $redirectLink){

-||-

}

frontPageEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

searchAllBooks(object $pdo, string $searchTerm){

preforms a SELECT query for books WHERE book\_titel = $searchTerm

returns books as an array

}

addFeturedItem(object $pdo){

preforms a INSERT query with the inputed data.

Returns TRUE if successful

Else return FALSE

}

editFeaturedItem(object $pdo, int $featItemId){

preforms an UPDATE query with the new data WHERE feat\_item\_id = $featItemId

return TRUE if successful

else return FALSE

}

deleteFeaturedItem(object $pdo, int $featItemId){

preforms a DELETE query on the books table WHERE feat\_item\_id = $featItemId

return TRUE if successful

else return FALSE

}

fetchFeturedItems(object $pdo, int $itemType){

prepares and executes SELECT query where item\_type = $itemType; to get the different featured items

return different arrays for different types of featured items

}

populateFeaturedItem(array $feturedItems){

runs foreach loop for the previous query results

returns TRUE upon completion

}

userEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addUser(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editUser(object $pdo, int $userId){

preforms an UPDATE query with the new data WHERE emp\_id = $userId

return TRUE if successful

else return FALSE

}

deleteUser(object $pdo, int $userId){

preforms a DELETE query on the books table WHERE emp\_id = $userId

return TRUE if successful

else return FALSE

}

fetchUsers(object $pdo, string $username){

prepares and executes SELECT query where emp\_username = $username;

return $userId and $userRole

}

populateUserField(array $users){

runs foreach loop for the previous query results

return TRUE upon completion

}

genreEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addGenre(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editGenre(object $pdo, int $genreId){

preforms an UPDATE query with the new data WHERE genre\_id= $genreId

return TRUE if successful

else return FALSE

}

deleteGenre(object $pdo, int $genreId){

preforms a DELETE query on the books table WHERE genre\_id= $genreId

return TRUE if successful

else return FALSE

}

fetchGenres(object $pdo, string $genreName){

prepares and executes SELECT query where genre\_name = $genreName;

return array

}

populateGenreField(array $genres){

runs foreach loop for the $genres array

return TRUE upon completion

}

publisherEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addPublisher(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editPublisher(object $pdo, int $pubId){

preforms an UPDATE query with the new data WHERE pub\_id = $pubId

return TRUE if successful

else return FALSE

}

deletePublisher(object $pdo, int $pubId){

preforms a DELETE query on the books table WHERE pub\_id = $pubId

return TRUE if successful

else return FALSE

}

fetchPublisher(object $pdo, string $pubName){

prepares and executes SELECT query where pub\_name = $pubName;

return $ publishers array

}

populatePublisherField(array $publishers){

runs foreach loop for the $publishers

return TRUE upon completion

}

authorEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addAuthor(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editAuthor(object $pdo, int $authorId){

preforms an UPDATE query with the new data WHERE author\_id = $auhorId

return TRUE if successful

else return FALSE

}

deleteAuthor(object $pdo, int $authorId){

preforms a DELETE query on the books table WHERE author\_id = $auhorId

return TRUE if successful

else return FALSE

}

fetchAuthors(object $pdo, string $authorName){

prepares and executes SELECT query where author\_name = $authorName;

return $authors array

}

populateAuthorField(array $authors){

runs foreach loop for the previous query results

returns true upon completion

}

illustratorEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addIllustrator(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editIllustrator(object $pdo, int $illustratorId){

preforms an UPDATE query with the new data WHERE illustrator\_id = $illustratorId

return TRUE if successful

else return FALSE

}

deleteIllustrator(object $pdo, int $illustratorId){

preforms a DELETE query on the books table WHERE illustrator\_id = $illustratorId

return TRUE if successful

else return FALSE

}

fetchIllustrators(object $pdo, int $illustratorName){

prepares and executes SELECT query where illustrator\_name = $illustratorName;

return $illustrators array

}

populateIllustratorField(array $illustrators){

runs foreach loop for the previous query results

return TRUE when completed

}

categoryEditor.php

checkUserRole(int $role, string $redirectLink){

compares int $role to $reqRole

returns true if $role > $reqRole

else redirects to $redirectLink

}

addCategory(object $pdo){

preforms a INSERT query with the inputed data.

return TRUE if successful

else return FALSE

}

editCategory(object $pdo, int $categoryId){

preforms an UPDATE query with the new data WHERE category\_id = $categoryId

return TRUE if successful

else return FALSE

}

deleteCategory(object $pdo, int $categoryId){

preforms a DELETE query on the books table WHERE category\_id = $categoryId

return TRUE if successful

else return FALSE

}

fetchCategorys(object $pdo, string $categoryName){

prepares and executes SELECT query where category\_name = $categoryName;

returns $categories array

}

populateCategoryField(array $categories){

runs foreach loop for the previous query results

return TRUE

}