# Funktionslista för login-system

checkUserRegisterInput($username, $email, $password, $passwordConfirm){

- Executed when the user submits the registration form

- Check if the entered username or email already exists in the database

- Verify that the passwords match

- Verify that the password is at least 8 characters long

- Verify that the email is valid

- Return true to proceed with the registration function

- If unsuccessful: Display error message on register.php

}

register($username, $email, $password){

- Encrypt the entered password using password\_hash(password, encryption method)

- Insert the entered information into the table\_user

- If successful: Redirect to login.php

- If not: Display error message on register.php

}

login($username\_email, $password){

- Executed when the user submits the login form

- Retrieve the user's information from the database where the username or email matches the input

- If no result: return false

- Compare the entered password with the saved password hash using password\_verify(entered password, hash retrieved from database)

- If the password is correct ->

- Create session variables with user info (user\_role, user\_id, user\_name)

- return true + redirect to home.php

- If not -> return false + display error message for the user on login.php

}

checkLoginStatus(){

- Check if the session variable, e.g., $\_SESSION['user\_id'], has a value

- If exists -> return true

- If not -> redirect to login.php

}

checkUserRole(user\_role\_value, comparison\_value){

- Check if the user's role value is greater than or equal to the comparison value

- If equal or greater: return true

- If not: return false

}

redirect($url){

- Move the user to a new URL

- Header(location: $url);

}

logout(){

- Unset the session array values

- Destroy the session

- Move the user to the login page

}

editUserInfo($userID, $username, $email, $password\_old, $password\_new, $password\_confirm, $firstname, $lastname(){

- Executed when the user submits the edit user form

- Check if the password has been updated

- Check that the passwords match

- Verify that the password is at least 8 characters long

- Encrypt the entered password using password\_hash(password, encryption method)

- Verify that the email is valid

- Update the entered information in table\_user

}

changeUserStatus($userid, $statusvalue){

- Update the user's status to $statusvalue

}

deleteUser($userid){

- Delete the user from the database, cascade linked values

}

searchUsers($input){

- Match $input with LIKE %$input% from the user table

- Return the results

- Loop through the list on the admin page

}

function fetchgenre($conn){

// Fetches all genre records from the table\_genre table

}

function fetchlanguage($conn){

// Fetches all language records from the table\_language table

}

function createbook($conn, $bookname, $bookauthor, $bookdescription, $bookillustrator, $bookage, $bookgenre, $booklanguage, $bookpages, $bookprice, $bookcreated, $image){

// Inserts a new book record into the table\_book table

}

function updatebook($conn, $bookname, $bookauthor, $bookdescription, $bookillustrator, $bookage, $bookgenre, $booklanguage, $bookpages, $bookprice, $bookcreated, $image, $book\_id){

// Updates an existing book record in the table\_book table

}

function selectbooks($conn){

// Selects all books from the table\_book table, along with their associated genre and language

}

function selectsinglebook($conn , $id){

// Selects a single book record from the table\_book table based on the provided book ID ($id), along with its associated genre and language

}

function selectSortedBooksPop($conn, $amount){

// Selects a limited number of books from the table\_book table sorted by popularity (genre ID 2)

}

function selectSortedBooksDesc($conn, $amount){

// Selects a limited number of books from the table\_book table sorted by creation date in descending order

}

function selectSortedBooksFeatured($conn){

// Selects all featured books from the table\_book table

}

function deleteBook($conn, $book\_id){

// Deletes a book record from the table\_book table based on the provided book ID ($book\_id)

}

function searchBooks($conn, $query) {

// Searches for books in the table\_book table based on the provided search query ($query), matching either the book name or author

}