

List of pages in application:

- index.php: main page with books searching and reservation functionality
- my_reservations.php: page which show all the reservations made by current user and the ability to remove reservations
- login.php: page for login in, users are redirected here if not logged in
- register.php: page for registering
- logout.php: page which loges users out and redirects to login page

Source code:

“constants.php”

```
<?php
$RESULTS_PER_PAGE = 5;
?>
```

“constants.php”

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbase = "books_reservations";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbase);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

“font_links.php”

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Roboto+Mono:ital,wght@0,100..700;1,100..700&family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
```

“footer.php”

```
<footer>
```

```
    Created by Konstantin Skliar
```

```
</footer>
```

“header.php”

```
<header>
  <div class="header-left">
    <h1><a href="index.php">Books reservations</a></h1>
    <a href="index.php">Search books</a>
    <a href="my_reservations.php">My reservations</a>
  </div>
  <div class="header-right">
    <?php
      if(isset($_SESSION["username"])) {
        echo '<a href="logout.php">Log out</a>';
      }
      else {
        echo '<a href="login.php">Login</a>';
        echo '<a href="register.php">Register</a>';
      }
    ?>
  </div>
</header>
```

“index.php”

```
<?php
session_start();
?>

<?php require "need_auth.php"; ?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Books reservations</title>

    <?php require "fonts_links.php"; ?>
    <link rel="stylesheet" href="styles.css" />
</head>
<body>
    <?php require "constants.php"; ?>
    <?php require "db_connect.php"; ?>

    <?php require "header.php"; ?>

    <?php
    // keeping search values persistent
    $title = isset($_GET["title"]) ? $_GET["title"] : "";
    $author = isset($_GET["author"]) ? $_GET["author"] : "";
    $category = isset($_GET["category"]) ? $_GET["category"] : -1;

    $page = (isset($_GET["page"]) && is_numeric($_GET["page"])) ? (int)($_GET["page"]) : 1;

    if($page < 1)
        $page = 1;

    // update page value according to button press (previous/next)
    if(isset($_GET["previous"]) && $page > 1)
        $page--;

    if(isset($_GET["next"]))
        $page++;

    ?>

    <?php
    function reserve_book() {
        global $conn;
        global $_POST;
        global $_SESSION;

        $isbn_esc = $conn->real_escape_string($_POST["isbn"]);
```

```

$username_esc = $conn->real_escape_string($_SESSION["username"]);

// check that book is not reserved
$sql = "SELECT * FROM books WHERE ISBN = '$isbn_esc' AND Reserved = 'Y'";
$result = $conn->query($sql);
if($result->num_rows > 0)
    return;

// setting the status of a book to reserved
$sql = "UPDATE books SET Reserved = 'Y' WHERE ISBN = '$isbn_esc'";
if ($conn->query($sql) === false) {
    echo "Error: " . $sql . "<br>" . $conn->error;
    return;
}

// inserting new reservation
$sql = "INSERT INTO reservations (ISBN, Username) VALUES ('$isbn_esc',
'$username_esc')";
if ($conn->query($sql) === false) {
    echo "Error: " . $sql . "<br>" . $conn->error;
    return;
}
}

// handling reserving a book
if(
    isset($_POST["isbn"]) &&
    isset($_POST["reserve_book"])
)
{
    reserve_book();
}
?>

<main>
    <form class="book-searchbar">
        <input name="title" placeholder="Book Title" value="<?php echo
htmlentities($title);?>"/>
        <input name="author" placeholder="Book Author" value="<?php echo
htmlentities($author);?>"/>
        <div>
            <label for="category">Category:</label>
            <select name="category" id="category">
                <option value="-1">All</option>
                <?php
                    // dynamically fetch categories from the database
                    $sql = "SELECT * FROM categories";
                    $result = $conn->query($sql);

```



```

        while($row = $result->fetch_assoc()) {
            echo '<option value="' . $row["CategoryID"] . "'';

            // select if it was selected before
            if($category == $row["CategoryID"])
                echo ' selected';

            echo '>' . $row["CategoryDescription"] . '</option>';
        }
    ?>
</select>
</div>
<input type="submit" value="Search" name="search" class="search-button"/>
</form>

<?php
// fetching search results
$title_esc = $conn->real_escape_string($title);
$author_esc = $conn->real_escape_string($author);
$category_esc = $conn->real_escape_string($category);

$sql_where_part = "WHERE BookTitle LIKE '%$title_esc%' and Author LIKE
'%$author_esc%'";
if($category != "-1")
    $sql_where_part = $sql_where_part . "AND b.Category = $category_esc";

// getting overall number of results
$sql = "SELECT count(*) as count FROM books b
    LEFT JOIN categories c ON b.Category = c.CategoryID
    " . $sql_where_part;

$result = $conn->query($sql);
if($result->num_rows === 0)
    die("Error retrieving data from the database: " . $conn->connect_error);

$row = $result->fetch_assoc();
$results_count = $row["count"];
$pages_count = (int)ceil($results_count / $RESULTS_PER_PAGE);

// getting the data itself
$results_skipped = ($page - 1) * $RESULTS_PER_PAGE;
$sql = "SELECT * FROM books b
    LEFT JOIN categories c ON b.Category = c.CategoryID" . " " . $sql_where_part . "
    " .

    "ORDER BY BookTitle ASC
    LIMIT $RESULTS_PER_PAGE OFFSET $results_skipped
    ";
$result = $conn->query($sql);

if ($result->num_rows > 0) {

```

```

// displaying results table
echo    '<div class="search-results">
        <table>
            <tr>
                <th>Book Title</th>
                <th>Author</th>
                <th>Edition</th>
                <th>Release Year</th>
                <th>Category</th>
                <th></th>
            </tr>';

// displaying data rows
while($row = $result->fetch_assoc()) {
    echo    '<tr>
            <td>' . htmlentities($row["BookTitle"]) . '</td>
            <td>' . htmlentities($row["Author"]) . '</td>
            <td>' . htmlentities($row["Edition"]) . '</td>
            <td>' . htmlentities($row["Year"]) . '</td>
            <td>' . htmlentities($row["CategoryDescription"]) . '</td>';

            if($row["Reserved"] == "N"){
                echo '<td><form method="POST">';
                echo    '<input name="isbn" value="' .
htmlentities($row["ISBN"]) . '" hidden/>';
                echo    '<input class="reserve-label" value="Reserve"
type="Submit" name="reserve_book" />';
                echo '</form></td>';
            }
            else {
                echo '<td><span class="reserved-label">Reserved</span></td>';
            }
        echo    '</tr>';
    }

echo    '</table>
        <form class="paging-bar">';
echo    '<input name="title" value="' . htmlentities($title) . '"
hidden/>';
echo    '<input name="author" value="' . htmlentities($author) . '"
hidden/>';
echo    '<input name="category" value="' . htmlentities($category) . '"
hidden/>';
echo    '<input name="page" value="' . htmlentities($page) . '"
hidden/>';

echo    '<div><input type="submit" value="Previous" name="previous";
if($page === 1)
    echo ' class="hide"';
echo'</div>';

```

```
echo          "<p>Page: $page / $pages_count</p>";

echo          '<div><input type="submit" value="Next" name="next"';
if($page === $pages_count)
    echo ' class="hide"';
echo' /></div>';

echo          '</form>
              </div>';
}
else {
    echo "<p>0 Results</p>";
}
?>
</main>

<?php require "footer.php"; ?>

<?php $conn->close() ?>
</body>
</html>
```

“login.php”

```
<?php
session_start();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>

    <?php require "fonts_links.php"; ?>
    <link rel="stylesheet" href="styles.css" />
</head>
<body>
    <?php require "db_connect.php"; ?>

    <?php require "header.php"; ?>

    <?php
    if(isset($_SESSION["username"]))
    {
        echo "
        <main>
            <h1>Login</h1>
            <p>Already logged in</p>
        </main>
        ";
        return 0;
    }
    ?>

    <?php
    // keeping entered information persistent
    $username = isset($_POST["username"]) ? $_POST["username"] : "";
    $password = isset($_POST["password"]) ? $_POST["password"] : "";

    $err_username = false;
    $err_password = false;

    function login() {
        global $username;
        global $password;

        global $err_username;
        global $err_password;

        global $conn;
```

```

// check that fields are not empty
if(strlen($username) == 0)
    return;
if(strlen($password) == 0)
    return;

$username_esc = $conn->real_escape_string($username);
$password_esc = $conn->real_escape_string($password);

// check that the user exists
$sql = "SELECT * from users WHERE Username = '$username_esc'";
$result = $conn->query($sql);
if($result->num_rows === 0) {
    $err_username = true;
    return;
}

$row = $result->fetch_assoc();
// check that password is right
if($password_esc !== $row["Password"]) {
    $err_password = true;
    return;
}

$_SESSION["username"] = $username_esc;
// redirect to main page
header("Location: index.php");
}

if(
    isset($_POST["username"]) &&
    isset($_POST["password"])
)
{
    login();
}
?>

<main>
    <h1>Login</h1>
    <form class="default-form" method="POST">
        <div class="form-field">
            <label for="username">Username</label>
            <input id="username" name="username" required value="<?php echo
htmlentities($username);?>" />
            <?php
            if($err_username)
                echo '<p class="form-field-error">No user with this username</p>';
            ?>

```

```
</div>

<div class="form-field">
    <label for="password">Password (6 characters)</label>
    <input type="password" id="password" name="password" minlength="6"
maxlength="6" required value="<?php echo htmlentities($password);?>" />
    <?php
    if($err_password)
        echo '<p class="form-field-error">Wrong password</p>';
    ?>
</div>

<input type="submit" value="Login">
</form>
</main>

<?php require "footer.php"; ?>

<?php $conn->close() ?>
</body>
</html>
```

“logout.php”

```
<?php
session_start();
session_destroy();
header("Location: index.php");
?>
```

“my_reservations.php”

```
<?php
session_start();
?>

<?php require "need_auth.php"; ?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My reservations</title>

    <?php require "fonts_links.php"; ?>
    <link rel="stylesheet" href="styles.css" />
</head>
<body>
    <?php require "constants.php"; ?>
    <?php require "db_connect.php"; ?>

    <?php require "header.php"; ?>

    <?php
function remove_reservations() {
    global $conn;
    global $_POST;
    global $_SESSION;

    $isbn_esc = $conn->real_escape_string($_POST["isbn"]);
    $username_esc = $conn->real_escape_string($_SESSION["username"]);

    // check that book is reserved
    $sql = "SELECT * FROM books WHERE ISBN = '$isbn_esc' AND Reserved = 'Y'";
    $result = $conn->query($sql);
    if($result->num_rows === 0)
        return;

    // setting the status of a book to not reserved
    $sql = "UPDATE books SET Reserved = 'N' WHERE ISBN = '$isbn_esc'";
    if ($conn->query($sql) === false) {
        echo "Error: " . $sql . "<br>" . $conn->error;
        return;
    }

    // deleting the reservation
    $sql = "DELETE FROM reservations WHERE ISBN = '$isbn_esc' AND Username = '$username_esc'";
    if ($conn->query($sql) === false) {
```



```

        echo "Error: " . $sql . "<br>" . $conn->error;
        return;
    }
}

// handling confirmation to remove a book from reservations
if(isset($_POST["conf_remove_book"]) && isset($_POST["isbn"])) {
    remove_reservations();
}
?>

<?php
// handling request to remove a book from reservations
if(isset($_POST["request_remove_book"]) && isset($_POST["isbn"])) {
    $isbn = $_POST["isbn"];
    $title = $_POST["title"];

    echo '
        <div class="tint-screen"></div>
        <form method="POST" class="confirm-removal-form">';
    echo '    <input name="isbn" value="' . htmlentities($isbn) . '" hidden/>';
    echo '    <p class="confirm-removal-form-title">Remove reservation for <span
class="book-title">\'' . htmlentities($title) . '\'/> ?</p>';
    echo '    <div class="options-wrapper">
        <div><input type="submit" value="Yes" class="yes-option"
name="conf_remove_book"/></div>
        <div><input type="submit" value="No" class="no-option"/></div>
    </div>
    </form>';
}

?>

<?php
$page = (isset($_GET["page"]) && is_numeric($_GET["page"])) ? (int)($_GET["page"]) : 1;

if($page < 1)
    $page = 1;

// update page value according to button press (previous/next)
if(isset($_GET["previous"]) && $page > 1)
    $page--;

if(isset($_GET["next"]))
    $page++;

?>

<main>
    <h1>Your reservations</h1>

```

```

<?php
// fetching reservations
$username_esc = $conn->real_escape_string($_SESSION["username"]);

$sql_main_part = "FROM reservations r
INNER JOIN books b USING(ISBN)
LEFT JOIN categories c ON b.Category = c.CategoryID
WHERE r.Username = '$username_esc'
ORDER BY BookTitle ASC";

// getting overall number of results
$sql = "SELECT count(*) as count " . $sql_main_part;
$result = $conn->query($sql);
if($result->num_rows === 0)
    die("Error retrieving data from the database: " . $conn->connect_error);

$row = $result->fetch_assoc();
$results_count = $row["count"];
$pages_count = (int)ceil($results_count / $RESULTS_PER_PAGE);

// getting the actual reservations according to the page
$results_skipped = ($page - 1) * $RESULTS_PER_PAGE;
$sql = "SELECT * " . $sql_main_part . " LIMIT $RESULTS_PER_PAGE OFFSET
$results_skipped";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // displaying results table
    echo    '<div class="search-results">
            <table>
            <tr>
                <th>Book Title</th>
                <th>Author</th>
                <th>Edition</th>
                <th>Release Year</th>
                <th>Category</th>
                <th></th>
            </tr>';

    // displaying data rows
    while($row = $result->fetch_assoc()) {
        echo    '<tr>
                <td>' . htmlentities($row["BookTitle"]) . '</td>
                <td>' . htmlentities($row["Author"]) . '</td>
                <td>' . htmlentities($row["Edition"]) . '</td>
                <td>' . htmlentities($row["Year"]) . '</td>
                <td>' . htmlentities($row["CategoryDescription"]) . '</td>';

        echo '<td><form method="POST">';
    }
}

```

```

                echo    '<input name="title" value="" .
htmlentities($row["BookTitle"]) . "" hidden/>';
                echo    '<input name="isbn" value="" . htmlentities($row["ISBN"])
. "" hidden/>';
                echo    '<input class="reserve-label" value="Remove"
type="Submit" name="request_remove_book" />';
                echo '</form/></td>';
            echo    '</tr>';
        }

        echo    '</table>
        <form class="paging-bar">';
        echo    '<input name="page" value="" . htmlentities($page) . ""
hidden/>';
        echo    '<div><input type="submit" value="Previous" name="previous"';
        if($page === 1)
            echo ' class="hide"';
        echo'</div>';

        echo    "&<p>Page: $page / $pages_count</p>";

        echo    '<div><input type="submit" value="Next" name="next"';
        if($page === $pages_count)
            echo ' class="hide"';
        echo'</div>';

        echo    '</form>
        </div>';
    }
    else {
        echo "<p>No reservations</p>";
    }
    ?>
</main>

<?php require "footer.php"; ?>

<?php $conn->close() ?>
</body>
</html>

```

“need_auth.php”

```
<?php
if(!isset($_SESSION["username"]))
    header("Location: login.php");
?>
```

“register.php”

```
<?php
session_start();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>

    <?php require "fonts_links.php"; ?>
    <link rel="stylesheet" href="styles.css" />
</head>
<body>
    <?php require "db_connect.php"; ?>

    <?php require "header.php"; ?>

    <?php
    if(isset($_SESSION["username"]))
    {
        echo "
        <main>
            <h1>Register</h1>
            <p>Already logged in</p>
        </main>
        ";
        return 0;
    }
    ?>

    <?php
    // keeping entered information persistent
    $username = isset($_POST["username"]) ? $_POST["username"] : "";
    $password = isset($_POST["password"]) ? $_POST["password"] : "";
    $conf_password = isset($_POST["conf_password"]) ? $_POST["conf_password"] : "";
    $firstname = isset($_POST["firstname"]) ? $_POST["firstname"] : "";
    $surname = isset($_POST["surname"]) ? $_POST["surname"] : "";
    $address_line_1 = isset($_POST["address_line_1"]) ? $_POST["address_line_1"] : "";
    $address_line_2 = isset($_POST["address_line_2"]) ? $_POST["address_line_2"] : "";
    $city = isset($_POST["city"]) ? $_POST["city"] : "";
    $telephone = isset($_POST["telephone"]) ? $_POST["telephone"] : "";
    $mobile = isset($_POST["mobile"]) ? $_POST["mobile"] : "";

    $err_username = false;
    $err_conf_password = false;
```

```

function register() {
    global $username;
    global $password;
    global $conf_password;
    global $firstname;
    global $surname;
    global $address_line_1;
    global $address_line_2;
    global $city;
    global $telephone;
    global $mobile;

    global $err_username;
    global $err_conf_password;

    global $conn;

    // check that fields are not empty
    if(strlen($username) == 0)
        return;
    if(strlen($password) == 0)
        return;
    if(strlen($conf_password) == 0)
        return;
    if(strlen($firstname) == 0)
        return;
    if(strlen($surname) == 0)
        return;
    if(strlen($address_line_1) == 0)
        return;
    if(strlen($address_line_2) == 0)
        return;
    if(strlen($city) == 0)
        return;
    if(strlen($telephone) == 0)
        return;
    if(strlen($mobile) == 0)
        return;

    // check that fields are proper length
    if(strlen($password) != 6)
        return;
    if(strlen($mobile) != 10)
        return;

    // check that password and confirmed password match
    if($password != $conf_password) {
        $err_conf_password = true;
        return;
    }
}

```

```

// check that telephone and mobile are numeric
if(!is_numeric($telephone))
    return;
if(!is_numeric($mobile))
    return;

$username_esc = $conn->real_escape_string($username);
$password_esc = $conn->real_escape_string($password);
$firstname_esc = $conn->real_escape_string($firstname);
$surname_esc = $conn->real_escape_string($surname);
$address_line_1_esc = $conn->real_escape_string($address_line_1);
$address_line_2_esc = $conn->real_escape_string($address_line_2);
$city_esc = $conn->real_escape_string($city);
$telephone_esc = $conn->real_escape_string($telephone);
$mobile_esc = $conn->real_escape_string($mobile);

// check that the username is not taken
$sql = "SELECT * from users WHERE Username = '$username_esc'";
$result = $conn->query($sql);
if($result->num_rows > 0) {
    $err_username = true;
    return;
}

// inserting new user
$sql = "INSERT INTO users (Username, Password, FirstName, Surname, AddressLine1,
AddressLine2, City, Telephone, Mobile)
    values ('$username_esc', '$password_esc', '$firstname_esc', '$surname_esc',
'$address_line_1_esc', '$address_line_2_esc',
'$city_esc', '$telephone_esc', '$mobile_esc')";

if ($conn->query($sql) === true) {
    $_SESSION["username"] = $username_esc;
    // redirect to main page
    header("Location: index.php");
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
}

if(
isset($_POST["username"]) &&
isset($_POST["password"]) &&
isset($_POST["conf_password"]) &&
isset($_POST["firstname"]) &&
isset($_POST["surname"]) &&
isset($_POST["address_line_1"]) &&
isset($_POST["address_line_2"]) &&
isset($_POST["city"]) &&

```

```

        isset($_POST["telephone"]) &&
        isset($_POST["mobile"])
    )
    {
        register();
    }
?>

<main>
    <h1>Register</h1>
    <form class="default-form" method="POST">
        <div class="form-field">
            <label for="username">Username</label>
            <input id="username" name="username" required value="<?php echo
htmlentities($username);?>"/>
            <?php
            if($err_username)
                echo '<p class="form-field-error">Username is already taken</p>';
            ?>
        </div>

        <div class="form-group">
            <div class="form-field">
                <label for="password">Password (6 characters)</label>
                <input type="password" id="password" name="password" minlength="6"
maxlength="6" required value="<?php echo htmlentities($password);?>"/>
            </div>
            <div class="form-field">
                <label for="conf_password">Confirm password</label>
                <input type="password" id="conf_password" name="conf_password"
minlength="6" maxlength="6" required value="<?php echo htmlentities($conf_password);?>"/>
                <?php
                if($err_conf_password)
                    echo '<p class="form-field-error">Passwords do not match</p>';
                ?>
            </div>
        </div>

        <div class="form-group">
            <div class="form-field">
                <label for="firstname">First name</label>
                <input id="firstname" name="firstname" required value="<?php echo
htmlentities($firstname);?>"/>
            </div>
            <div class="form-field">
                <label for="surname">Surname</label>
                <input id="surname" name="surname" required value="<?php echo
htmlentities($surname);?>"/>
            </div>
        </div>
    </form>

```



```

        <div class="form-group">
            <div class="form-field">
                <label for="address_line_1">Address line 1</label>
                <input id="address_line_1" name="address_line_1" required value="<?php
echo htmlentities($address_line_1);?>" />
            </div>
            <div class="form-field">
                <label for="address_line_2">Address line 2</label>
                <input id="address_line_2" name="address_line_2" required value="<?php
echo htmlentities($address_line_2);?>" />
            </div>
            <div class="form-field">
                <label for="city">City</label>
                <input id="city" name="city" required value="<?php echo
htmlentities($city);?>" />
            </div>
        </div>

        <div class="form-group">
            <div class="form-field">
                <label for="telephone">Telephone</label>
                <input type="number" id="telephone" name="telephone" required
value="<?php echo htmlentities($telephone);?>" />
            </div>
            <div class="form-field">
                <label for="mobile">Mobile (10 digits)</label>
                <input type="text" id="mobile" name="mobile" pattern="\d*" minlength="10"
maxlength="10" required value="<?php echo htmlentities($mobile);?>" />
            </div>
        </div>

        <input type="submit" value="Register">
    </form>
</main>

<?php require "footer.php"; ?>

<?php $conn->close() ?>
</body>
</html>

```

“styles.php”

```
/* ----- */
/* Custom properties */
/* ----- */

:root {
  /* font-sizes */
  --fs-100: 1.1rem;
  --fs-200: 1.2rem;
  --fs-300: 1.3rem;
  --fs-350: 1.4rem;
  --fs-400: 1.6rem;
  --fs-500: 1.7rem;
  --fs-600: 1.8rem;
  --fs-800: 2rem;
  --fs-850: 2.5rem;
  --fs-900: 3rem;

  /* font-families */
  --ff-sans: "Roboto", sans-serif;
  --ff-mono: "Roboto Mono", monospace;

  /* colors */
  --c-light: #fcfcfc;

  --c-main: #2f4858;
  --c-main-light: #4d748d;

  --c-dark: #000000;
  --c-dark-100: #f6f6f6;
  --c-dark-200: #e7e7e7;
  --c-dark-300: #c0c5c9;
  --c-dark-500: #84878a;

  --c-error: #eb2727;

  --br-400: 0.4rem;
  --br-600: 1rem;
}

/* ----- */
/* Resets */
/* ----- */

*,
*::before,
*::after {
  box-sizing: border-box;
}
```

```
body,
h1,
h2,
h3,
h4,
h5,
p,
section,
img {
    margin: 0;
    padding: 0;
    font-weight: inherit;
    font-size: inherit;
    font-family: inherit;
}

body {
    line-height: 1.5;
    min-height: 100vh;
    background-color: hsl(0, 0%, 100%);
}

input,
button,
textarea,
select {
    font: inherit;
    padding: 0;
    margin: 0;
    border: none;
    color: inherit;
}

input {
    outline: none;
}

a {
    text-decoration: none;
    color: inherit;
    cursor: pointer;
}

ul {
    list-style: none;
    padding: 0;
    margin: 0;
}
```

```
img {
  width: 100%;
  display: block;
}

/* rem scaling */
html {
  font-size: 62.5%;
}

/* ----- */
/* Main styles      */
/* ----- */

body {
  font-size: var(--fs-400);
  font-family: var(--ff-sans);
  background-color: var(--c-light);
  color: var(--c-dark);

  display: flex;
  flex-direction: column;
}

header {
  display: flex;
  justify-content: space-between;
  padding: 1rem;
  padding-left: 2rem;
  padding-right: 7rem;
  margin-bottom: 3rem;
  letter-spacing: 0.05rem;
}

header a {
  text-decoration: underline;
}

h1 a {
  text-decoration: none;
}

.header-left {
  display: flex;
  align-items: center;
  gap: 3rem;
}

.header-right {
  display: flex;
}
```

```
    align-items: center;
    gap: 2rem;
}

h1 {
    font-size: var(--fs-900);
    font-weight: 700;
    margin-bottom: 1rem;
}

main {
    display: flex;
    flex-direction: column;
    align-items: center;
    flex-grow: 1;
}

.book-searchbar {
    display: flex;
    align-items: center;
    gap: 2rem;
    margin-bottom: 3rem;
    font-size: var(--fs-400);
}

.book-searchbar label {
    margin-right: 0.5rem;
    margin-left: 1rem;
    font-size: var(--fs-600);
    text-align: right;
}

input,
select {
    border: 1px solid var(--c-dark);
    border-radius: var(--br-400);
    padding: 0.8rem;
}

input[type="submit"] {
    background-color: var(--c-main);
    color: var(--c-light);
    padding: 0.8rem;
    padding-left: 1.5rem;
    padding-right: 1.5rem;
    border-radius: var(--br-400);
    border: none;
}

input[type="submit"]:hover {
```

```
    background-color: var(--c-main-light);
}

input[type="submit"]:disabled {
    color: var(--c-dark-300);
}

input[type="submit"]:hover:disabled {
    background-color: var(--c-main);
}

.search-button {
    margin-left: 3rem;
}

.search-results {
    min-width: 80rem;
    border-radius: var(--br-600);
    overflow: hidden;
    --border: 1px solid var(--c-dark-300);
}

.search-results table {
    width: 100%;
    border-collapse: collapse;
}

.search-results table th {
    padding-top: 1.2rem;
    padding-bottom: 1.2rem;
    text-align: left;
    background-color: var(--c-main);
    color: var(--c-light);
    border-top: none;
}

.search-results table td,
.search-results table th {
    border-left: var(--border);
    border-top: var(--border);
    padding: 0.8rem;
}

.search-results table td:first-child,
.search-results table th:first-child {
    border-left: none;
}

.search-results table tr:nth-child(even) {
    background-color: var(--c-dark-200);
}
```

```

}

.paging-bar {
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 3rem;
  background-color: var(--c-main);
  color: var(--c-light);
  border-top: var(--border);
  padding: 0.8rem;
}

.paging-bar input[type="submit"] {
  padding: 0;
  padding-left: 1rem;
  padding-right: 1rem;
}

.paging-bar > * {
  width: 10rem;
  text-align: center;
}

.reserved-label,
.reserve-label {
  display: inline-block;
  width: 100%;
  height: 100%;
  padding: 0.5rem;
  text-align: center;
}

.reserve-label {
  border-radius: var(--br-400);
}

.reserve-label {
  background-color: var(--c-main);
  color: var(--c-light);
  padding: 0.5rem;
  border-radius: var(--br-400);
}

.reserve-label:hover {
  background-color: var(--c-main-light);
}

footer {
  display: flex;

```

```
    justify-content: center;
    font-size: var(--fs-300);
    color: var(--c-dark-500);
    padding: 0.2rem;
    margin-top: 5rem;
}

.hide {
    visibility: hidden;
}

.default-form {
    display: flex;
    flex-direction: column;
    gap: 3rem;
    min-width: 40rem;
}

.form-group {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
    width: 100%;
}

.form-field {
    display: flex;
    flex-direction: column;
}

.form-field-error {
    color: var(--c-error);
    font-weight: 600;
}

.tint-screen {
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;

    background-color: rgba(0, 0, 0, 0.2);
}

.confirm-removal-form {
    position: absolute;
    left: 50%;
    top: 50%;
    transform: translate(-50%, -50%);
```



```
    background-color: white;
    padding: 2rem;
    z-index: 10;
    border-radius: var(--br-600);
}

.confirm-removal-form-title {
    font-size: var(--fs-600);
    margin-bottom: 2rem;
}

.confirm-removal-form-title .book-title {
    font-weight: 600;
}

.confirm-removal-form .options-wrapper {
    display: flex;
    gap: 6rem;
    justify-content: center;
}

.confirm-removal-form .options-wrapper > * {
    width: 6rem;
}

.confirm-removal-form .options-wrapper input {
    padding: 0.5rem;
    width: 100%;
    height: 100%;
}
```

Screens of functionality:

1) Login

Books reservations

[Search books](#)

[My reservations](#)

[Login](#)

[Register](#)

Login

Username

user1

Password (6 characters)

.....

Login

2) Register

Books reservations

[Search books](#)

[My reservations](#)

[Login](#)

[Register](#)

Register

Username

user2

Password (6 characters)

.....

Confirm password

.....

First name

First

Surname

Second

Address line 1

Line 1

Address line 2

Line 2

City

Dublin

Telephone

9999999999

Mobile (10 digits)

9999999999

Register

3) Search for a book and reserve

Books reservations[Search books](#)[My reservations](#)[Log out](#)

Category:

All

Search

Book Title	Author	Edition	Release Year	Category	
DaVinci Code	Dan Brown	1	2003	Fiction	Reserve
Exploring Peru	Stephanie Birchie	4	2005	Travel	Reserved
How to cook Italian food	Jamie Oliver	2	2005	Cookery	Reserved
My life in bits	John Smith	1	2001	Health	Reserved
My life in picture	Kevin Graham	8	2004	Health	Reserved

Previous

Page: 2 / 3

Next

Created by Konstantin Skliar

With some filters applied:

Books reservations[Search books](#)[My reservations](#)[Log out](#)

Category:

Health

Search

Book Title	Author	Edition	Release Year	Category	
A guide to nutrition	John Thorpe	2	1997	Health	Reserved
My life in bits	John Smith	1	2001	Health	Reserved
Shooting History	Jon Show	1	2003	Health	Reserved

Page: 1 / 1

Created by Konstantin Skliar

4) View reserved books

Books reservations

[Search books](#)[My reservations](#)[Log out](#)

Your reservations

Book Title	Author	Edition	Release Year	Category	
Computers for idiots	Susan O'Neill	5	1998	Technology	<button>Remove</button>
Computers in Business	Alicia Oneill	3	1997	Biography	<button>Remove</button>
Cooking for children	Anabelle Sharpe	1	2003	Cookery	<button>Remove</button>
Exploring Peru	Stephanie Birchie	4	2005	Travel	<button>Remove</button>
How to cook Italian food	Jamie Oliver	2	2005	Cookery	<button>Remove</button>
Page: 1 / 2 Next					

5) Remove reserved books

After clicking 'remove' the confirmation box appears:

Books reservations [Search books](#) [My reservations](#) [Log out](#)

Your reservations

Book Title	Author	Edition	Release Year	Category	
Computers for idiots	Susan O'Neill	5	1998	Technology	<button>Remove</button>
Computers in Business	Alicia Oneill	3	1997	Biography	<button>Remove</button>
Cooking for children				Cookery	<button>Remove</button>
Exploring Peru				Travel	<button>Remove</button>
How to cook Italian food	Jamie Oliver	2	2005	Cookery	<button>Remove</button>

Remove reservation for 'Computers for idiots' ?
Yes No

Page: 1 / 2 [Next](#)

Created by Konstantin Skliar