**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 = "";

$database = "books\_reservations";

// Create connection

$conn = new mysqli($servername, $username, $password, $database);

// 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) . '\'</span> ?</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>

            <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;

}

.reserved-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**

A screenshot of a login screen

Description automatically generated

1. **Register**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Search for a book and reserve**

A screenshot of a computer

Description automatically generated

With some filters applied:

A screenshot of a computer

Description automatically generated

1. **View reserved books**

A screenshot of a web page

Description automatically generated

1. **Remove reserved books**

After clicking ‘remove’ the confirmation box appears:

A screenshot of a web page

Description automatically generated