Internet Programming Project Report

Subject: CSE341 (Internet Programming)

Submitted to:

Dr. Hesham Farag

TA. Eslam Sayed Abd El Gaber

18P2491 Mazen Mostafa Fawzy Moussa

18P7232 Mohamed Mokhtar Abdelrasoul

18P6000 Mostafa Lotfy Mostafa Shehata

18P6713 Youssef Maher Nader Halim

Group: 2

Section: 2

Due Date: 20/6/2022

Table of Contents

Online Library Management System

Basic Assumptions

The library system has two types of users, regular users (which are the patrons of the library), and administrative staff (which include librarians and any other administrative staff that need access to the website). Some web pages have a different view based on the logged-in user, for example, the Book Details page will have a “Remove book” button for administrative users to be able to remove a book. The website is initialized with 25 books and 10 accounts (both admin and member), these initializations are stored on the localStorage when the website is first booted and are updated as needed.

Stored Accounts

There are 10 accounts stored by default on the website. The accounts are stored in the local storage, and they can be retrieved, and new accounts can be added to it.

Email	Password	Role
admin@admin.com	admin	Admin
bookkeeper@library.com	bookkeeper	Admin
ymaher@912.com	YMAHER99_DOTa2@su	Member
mazen@mazen.com	Mazeeeeeen	Member
lotfy@lotfy.com	LOTFYYYYY	Member
mokh@gmail.com	m0khtar222	Member
omar@yahoo.com	olt0fy	Member
unknown@hotmail.com	IamHIDD3N	Member
jdoe@gmail.com	doeD0Edoe	Member
ahmed12@gmail.com	aHaz3m223	Member

Accessing Website

There are three methods to access the website

- Run the index.html file.
- Run as a Live Server from VSCode.
- Run through the deployment on GitPages (found here https://mazen-ghaleb.github.io/OnlineLibrary/index.html)

GitHub Repo

Here is the GitHub repository link https://github.com/Mazen-Ghaleb/OnlineLibrary .

Common Elements Between Pages

Every page has a navigation bar at the top to allow the user to navigate the different pages found on the website. The navigation bar will change based on if a user is logged in or not, displaying the user's profile picture and adding a sign-out button, and if the logged-in account has the "Admin" role the navbar will include links to the "Add Book" and "Browse Members" pages which are admin only pages. Additionally, the navigation bar will normally have a link to the homepage and will have a search bar where users can search for books using the book's title or the author's name. The navigation bar uses bootstrap's navbar components. Each page contains a search bar at the top to enable the user to search for books.

Pages

For our project, we chose a design that contains the following web pages:

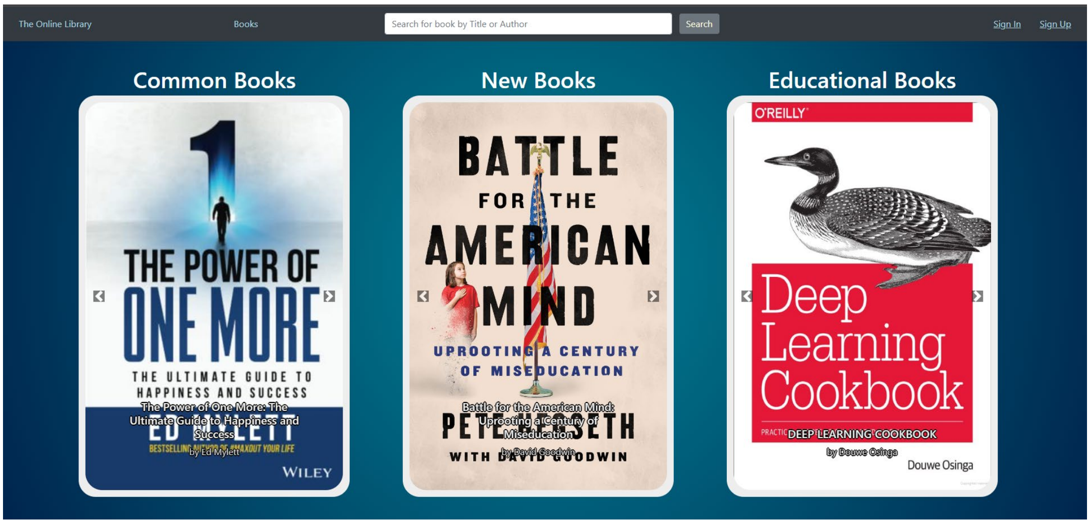
1. Homepage
2. Book Details
3. Profile page (Member Details)
4. Browse Books (Used to search for books, search terms can be matched to
5. Browse All Books (All books)
6. Add Book (Admin Only)
7. Browse Members (Admin Only)
8. Sign In
9. Sign Up
10. Password Reset

1) Homepage (index.html)

Description

The homepage contains multiple categories of books, namely common books, educational books, and new books, for each of these categories a carousel demonstrates the books available within these categories. The user can press on a book's image to navigate to the book's book details page.

Screenshots


The Online Library
Books
Search for book by Title or Author
Search
Sign In
Sign Up
Common Books
New Books
Educational Books
The Power of One More: The Ultimate Guide to Happiness and Success
by Ed Mylett
Battle for the American Mind: Uprooting a Century of Miseducation
by David Goodwin
DEEP LEARNING COOKBOOK
by Douwe Osinga

Code Description

In the index.html page we include the scripts and stylesheets associated with the homepage and the ones required for bootstrap components. We then run a script that generates the navigation bar and displays account details within it. In the body of the html file, we create three empty carousels from bootstrap that will be dynamically filled with books from each category, the carousels are filled using a script found in BookData.js.

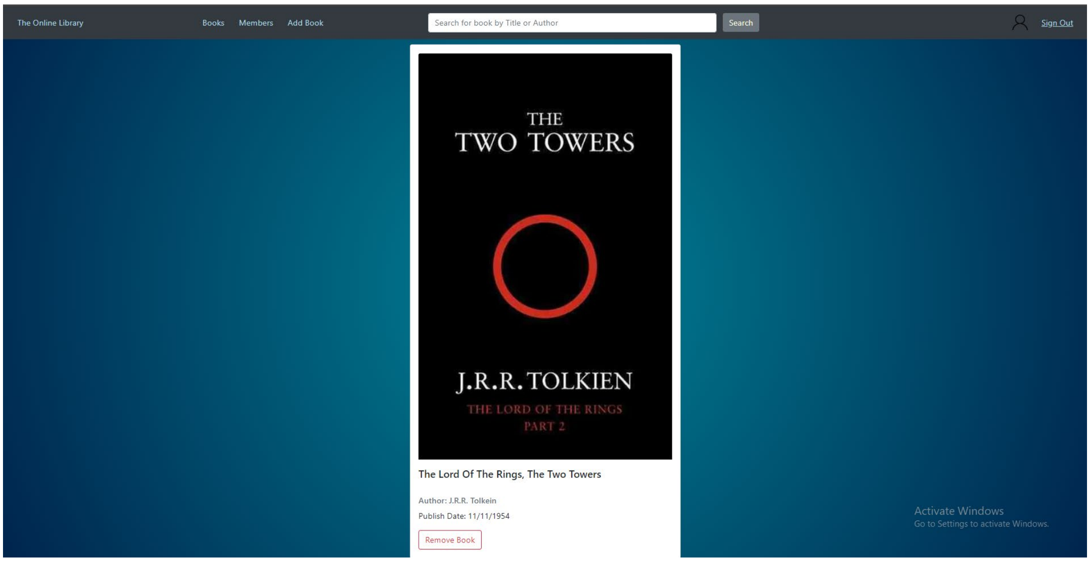
The function "createCarousel" is given an array of books to fill the carousel from and the number of the carousel to fill. The function then dynamically creates the elements needed to fill the carousel (e.g. "img" for the image and an "h5" for the book title, etc.) and appends them to the correct position in the HTML file. The array given to the function is generated from the "loadBooks" function found in BookData.js which populates the array from an array called AllBooks, which is found localStorage.

2) Book Details (bookDetails.html)

Description

This page contains the details of a certain book within the library, the details include the book’s cover, its title, its author’s name, and its date of publishing. If the logged-in user has the role “Admin” then below the book’s details a “Remove Book” button will be visible that allows the admin to remove the book from the library.

Screenshots

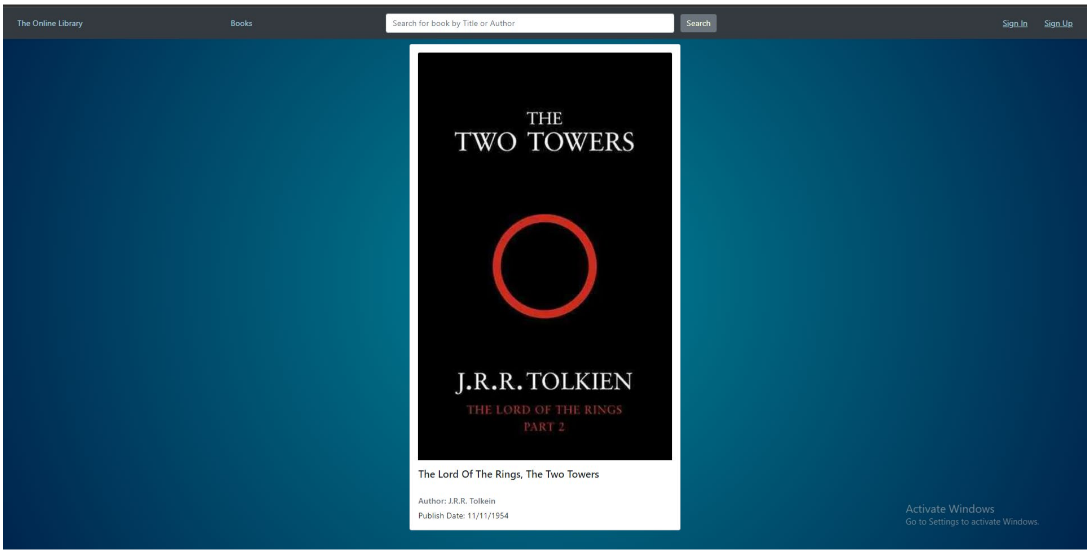


The Lord Of The Rings, The Two Towers

Author: J.R.R. Tolkein

Publish Date: 11/11/1954

The book details page when logged in as an admin.



The Lord Of The Rings, The Two Towers

Author: J.R.R. Tolkein

Publish Date: 11/11/1954

The book details page when logged in as a normal member.

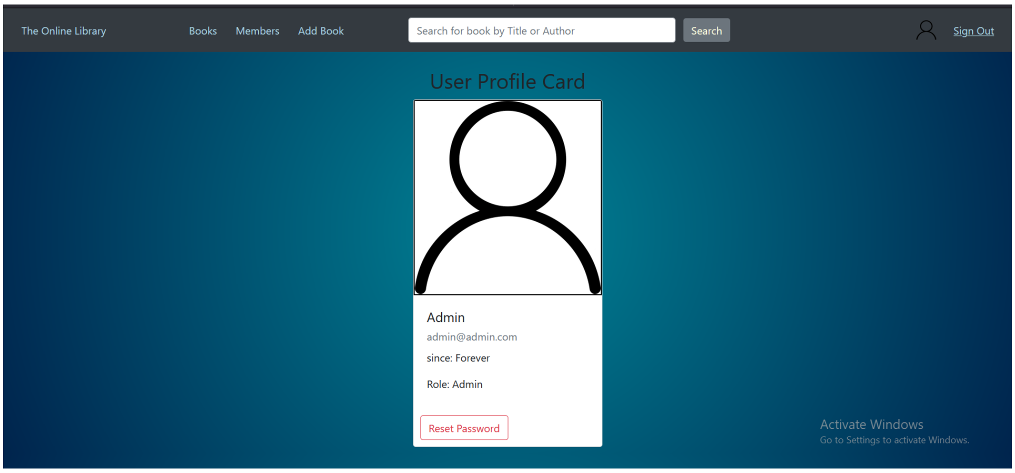
Code Description

This page is opened when a user clicks the image of a book cover on any of the pages containing the books. When the image is clicked, the user is redirected to the book details page while appending to the URL a variable called "book" with the value being the title of the book that was pressed. Once on the Book details page, we generate the navigation bar as per usual and create an empty card to be filled by the function "displayBookDetails" from BookData.js. The aforementioned function grabs the URL of the book details page in order to get the title of the selected book then grabs the empty card generated and fills it with the HTML required to display the details of the book in the card.

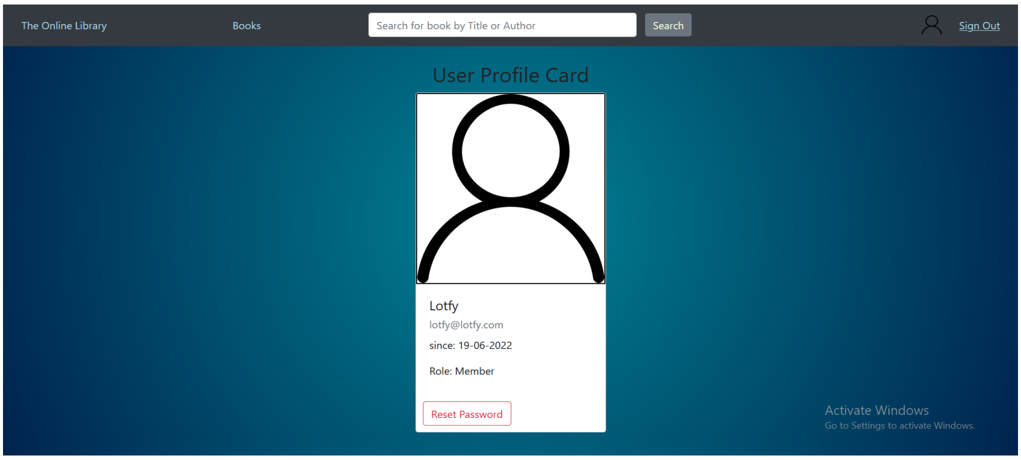
3) Profile Page (Profile.html)

This page will allow a user to view his profile. The profile will contain the username, email address, sign-up date and his role on the website.

Screenshots



Admin
admin@admin.com
since: Forever
Role: Admin



Lotfy
lotfy@lotfy.com
since: 19-06-2022
Role: Member

Code Description

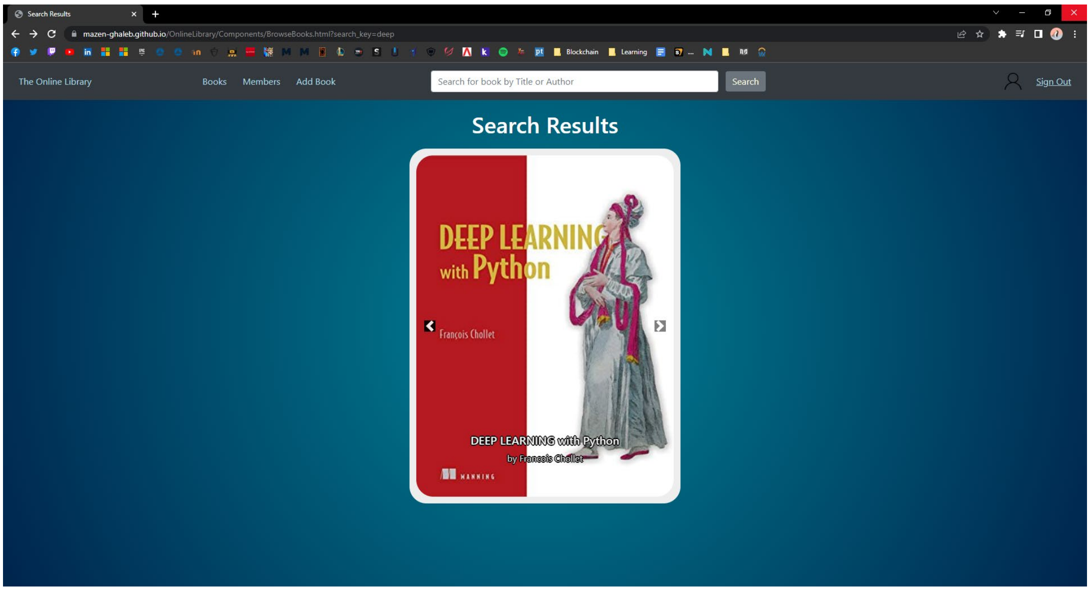
In the Profile.html page we include the scripts and stylesheets associated with the profile page and the ones required for bootstrap components. We then run a script that generates the navbar and displays account details within it.

This page is visited when the user presses on the profile picture in the navigation bar or when an admin selects a member from the browse members page. This page is used to display the information of the user. The Email of the logged-in user is passed as a parameter in the URL and then is used to retrieve the user data from local storage. The data is then displayed in the form shown in the previous screenshots. A button is shown to allow the user to reset his password. The card component from bootstrap is used and is populated using JavaScript code.

4) Browse Books (BrowseBooks.html)

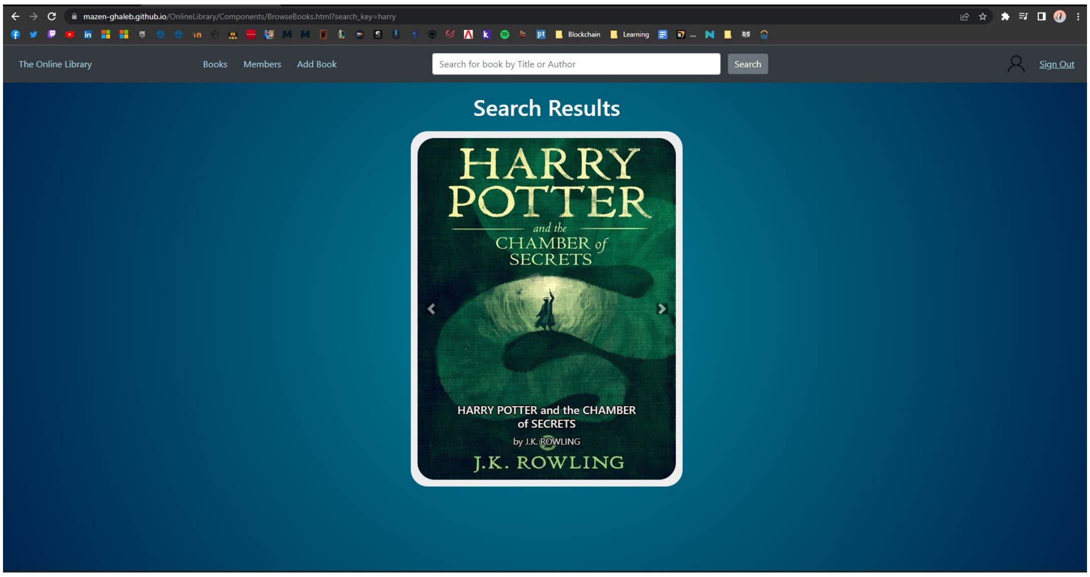
Users will be redirected from any page according to the search bar to this page and the page will be filled according to the search term, to correspond with their query. The page shows a Carousel demonstrating books from any category that match the term in its title or in the author's name. The user can press on a book's image to navigate to the book's book details page.

Screenshots



Search Results
mazen-ghaleb.github.io/OnlineLibrary/Components/BrowseBooks.html?search_key=deep
Blockchain
Learning
The Online Library
Books
Members
Add Book
Search for book by Title or Author
Search
Sign Out
Search Results
DEEP LEARNING with Python
François Chollet
DEEP LEARNING with Python
by François Chollet
MANNING

After entering 'Deep' in the search bar.


mazen-ghaleb.github.io/OnlineLibrary/Components/BrowseBooks.html?search_key=harry
Blockchain
Learning
The Online Library
Books
Members
Add Book
Search for book by Title or Author
Search
Sign Out
Search Results
HARRY POTTER and the CHAMBER of SECRETS
HARRY POTTER and the CHAMBER of SECRETS
by J.K. ROWLING
J.K. ROWLING

After entering 'Harry' in the search bar.

Code Description

On the BrowseBooks.html page we include the scripts and stylesheets associated with the page and the ones required for bootstrap components like the navigation bar. We then run a script that generates the navigation bar and displays account details within it. In the body of the HTML file, we create one empty carousel from bootstrap that will be dynamically filled with books resulting from the search, the carousel is filled using a script found in BookData.js.

The function "browseSearchResults" is a function that accesses the search results and then dynamically creates the elements needed to fill the carousel (e.g. "img" for the image and an "h5" for the book title, etc.) and appends them to the correct position in the HTML file.

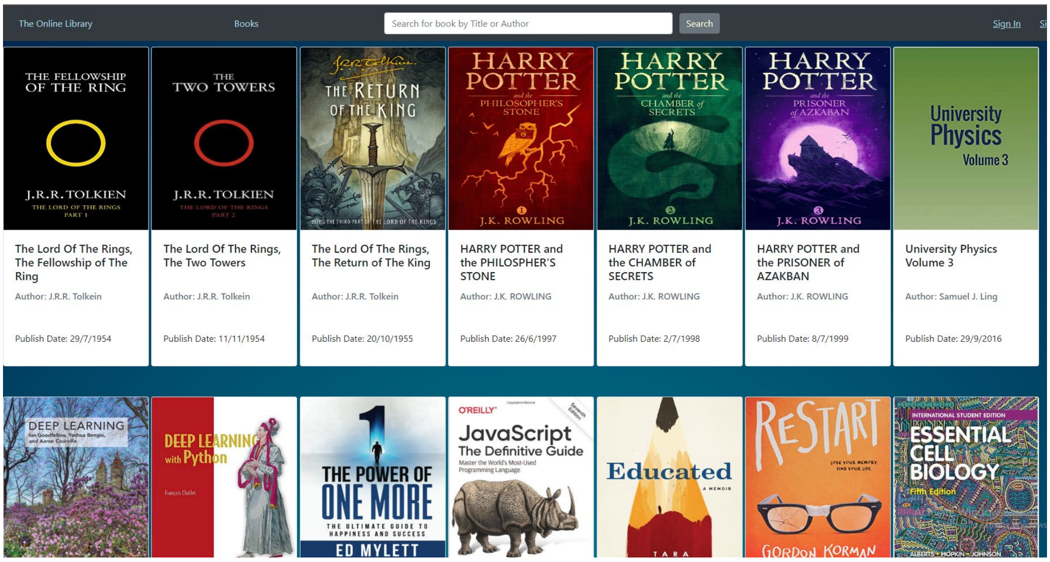
The search results array is initially stored in the localStorage whenever a search bar is used by calling the function "search_books" in the index.js, which loops around all books regardless of the category and adds the books that match the search term.

5) Browse All Books (browseAllBooks.html)

Description

Users can open this page through the “books” link in the navigation bar to view all the books currently available in the library. Additionally, the book’s image can be clicked to redirect the user to the book’s book details page.

Screenshots



The Lord Of The Rings, The Fellowship of The Ring
Author: J.R.R. Tolkein
Publish Date: 29/7/1954

The Lord Of The Rings, The Two Towers
Author: J.R.R. Tolkein
Publish Date: 11/11/1954

The Lord Of The Rings, The Return of The King
Author: J.R.R. Tolkein
Publish Date: 20/10/1955

HARRY POTTER and the PHILOSPHER'S STONE
Author: J.K. ROWLING
Publish Date: 26/6/1997

HARRY POTTER and the CHAMBER of SECRETS
Author: J.K. ROWLING
Publish Date: 2/7/1998

HARRY POTTER and the PRISONER of AZAKBAN
Author: J.K. ROWLING
Publish Date: 8/7/1999

University Physics Volume 3
Author: Samuel J. Ling
Publish Date: 29/9/2016

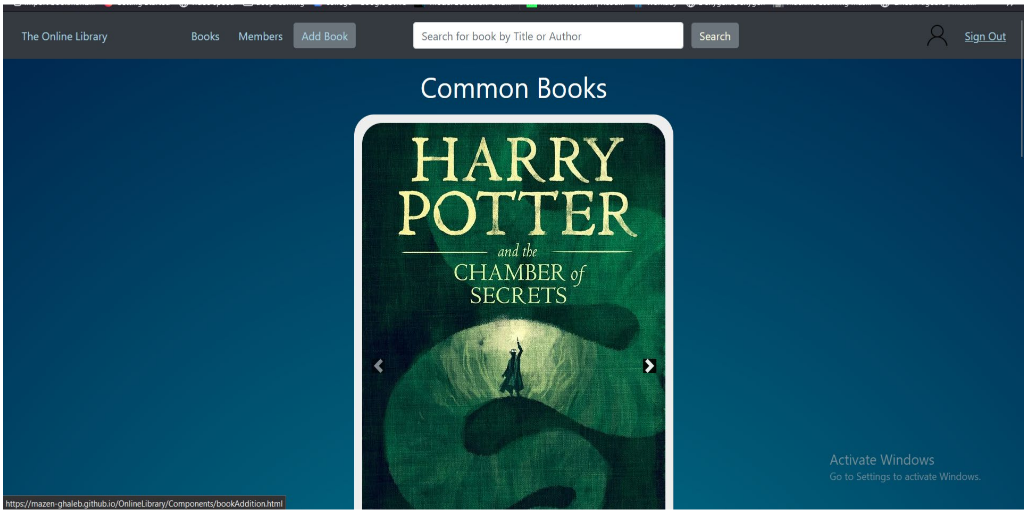
Code Description

The browseAllBooks.html file is a simple file that simply includes the scripts from index.js and BookData.js. The “generateNavBar” and “Display_account_data” functions found in index.js are then called at the beginning of the body, then we create a div that will hold the cards generated for each book. We then call the “generateCardsForAllBooks” function from BookData.js. The aforementioned function fetches the previously mentioned div that will now hold the cards and appends the HTML needed to generate each book’s card to the div, this is done by looping on an array of all books, named “AllBooks”, which is found in localStorage.

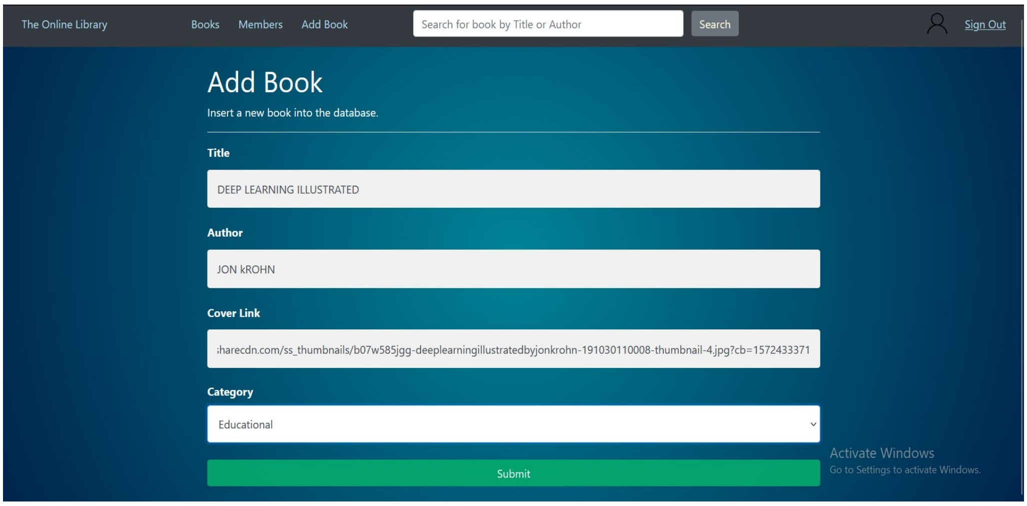
6) Add Book (bookAddition.html)

This page will allow a user with an admin role to add new books to the website. It can be accessed from the navigation bar when an admin is logged in.

Screenshots


The Online Library
Books
Members
Add Book
Search for book by Title or Author
Search
Sign Out
Common Books
HARRY POTTER and the CHAMBER of SECRETS
Activate Windows
Go to Settings to activate Windows.
https://mazen-ghaleb.github.io/OnlineLibrary/Components/bookAddition.html

Pressing Add Book when logged as an admin



Add Book

Insert a new book into the database.

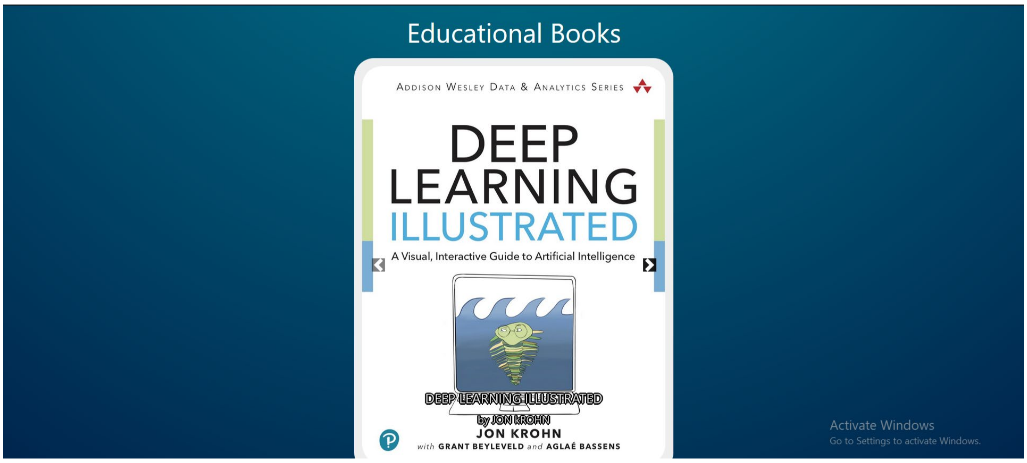
Title: DEEP LEARNING ILLUSTRATED

Author: JON kROHN

Cover Link: ;harecdn.com/ss_thumbnails/b07w585jgg-deeplearningillustratedbyjonkrohn-191030110008-thumbnail-4.jpg?cb=1572433371

Category: Educational

Filling the form to add a new book


Educational Books
Addison Wesley Data & Analytics Series
DEEP LEARNING ILLUSTRATED
A Visual, Interactive Guide to Artificial Intelligence
JON KROHN
with GRANT BEYLEVELD and AGLAÉ BASSENS

The new Added book is displayed on the home page

Code Description

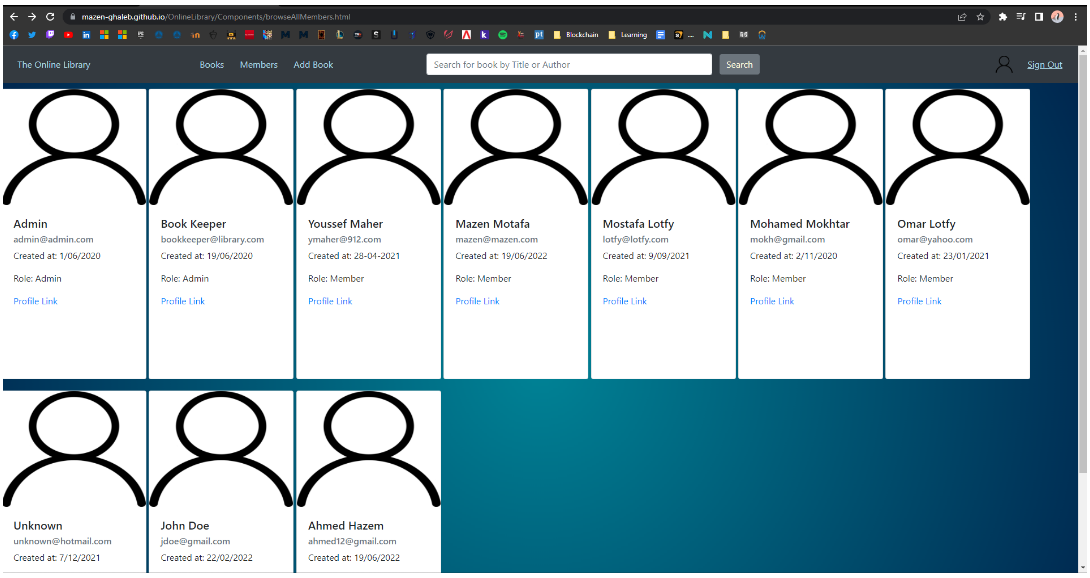
On the bookAddition.html page, we include the scripts and stylesheets associated with the book addition and the ones required for bootstrap components. We then run a script that generates the navigation bar and displays account details within it.

This page is only available to users with admin roles. The page is visited when the Add Book button is pressed. The button is also visible to only users with admin roles. The user is then presented with a form that allows them to enter book data. The book data includes title, author, category, date and a URL to the cover of the book. After submitting the form, the data of the new book is added to the localStorage using the “add_book” function found in “BookData.js”.

7) Browse Members (browseAllMembers.html)

This page will enable admins to see all members on the page including their details such as the email, name, creation date, role, and profile links.

Screenshots


mazen-ghaleb.github.io/OnlineLibrary/Components/browseAllMembers.html
Blockchain
Learning
The Online Library
Books
Members
Add Book
Search for book by Title or Author
Search
Sign Out

Admin
admin@admin.com
Created at: 1/06/2020
Role: Admin
Profile Link

Book Keeper
bookkeeper@library.com
Created at: 19/06/2020
Role: Admin
Profile Link

Youssef Maher
ymaher@912.com
Created at: 28-04-2021
Role: Member
Profile Link

Mazen Motafa
mazen@mazen.com
Created at: 19/06/2022
Role: Member
Profile Link

Mostafa Lotfy
lotfy@lotfy.com
Created at: 9/09/2021
Role: Member
Profile Link

Mohamed Mokhtar
mokh@gmail.com
Created at: 2/11/2020
Role: Member
Profile Link

Omar Lotfy
omar@yahoo.com
Created at: 23/01/2021
Role: Member
Profile Link

Unknown
unknown@hotmail.com
Created at: 7/12/2021

John Doe
jdoe@gmail.com
Created at: 22/02/2022

Ahmed Hazem
ahmed12@gmail.com
Created at: 19/06/2022

Code Description

In browseAllMembers.html, the page includes “index.js” for general functionality, such as generating the navigation bar. Additionally, it includes AccountData.js to use functions and manipulate account data. This page is made for only admin usage and therefore before loading any element the page will make sure that the account is logged and that account has admin role, else it will return to the previous page without even loading for security. If an admin accesses the page, then the page will generate cards for all members in the system.

The function “generateCardsForAllMembers” in AccountData.js, dynamically creates cards for each account in the stored accounts array found in localStorage and therefore creates cards for all accounts.

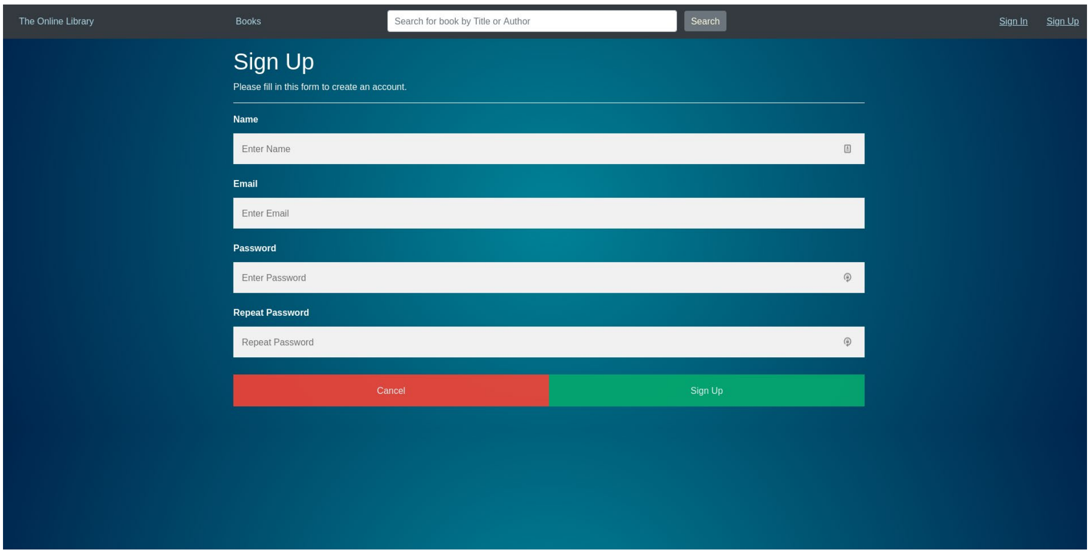
8) Sign Up (SignUp.html)

Description

If a user does not have an account, they can easily create one by clicking the “Sign Up” button on the top-right of the screen. This redirects the user to another page where they can enter their new account details. The user can press the “Cancel” button in order to go back to the previous page, if preferred.

If the user tries to create an account using an email that is already associated with another account, they get a note stating that this email cannot be used. If they fail to enter the password exactly the same in the “Repeat Password” field, they get another note stating that the password was not repeated correctly. However, if all details are validated, the account is created, and the user is redirected to the “Sign In” page.

Screenshots



Sign Up

Please fill in this form to create an account.

Name

Enter Name

Email

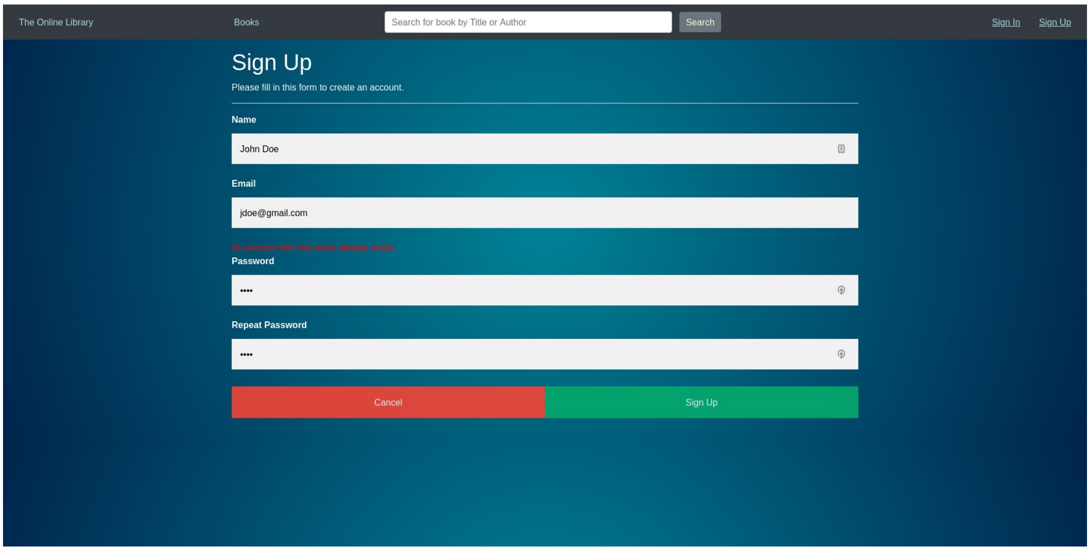
Enter Email

Password

Enter Password

Repeat Password

Repeat Password



Sign Up

Please fill in this form to create an account.

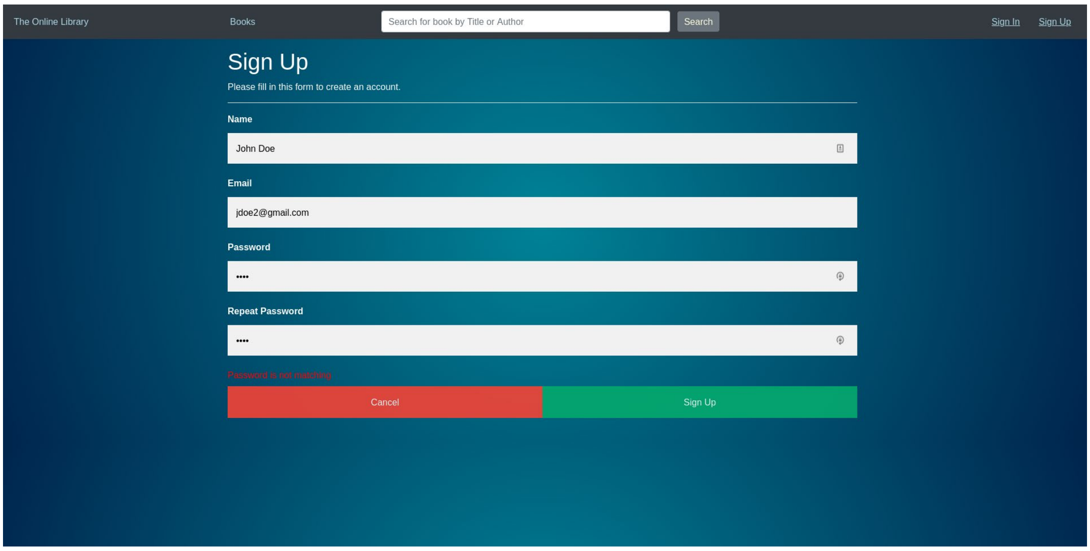
Name: John Doe

Email: jdoe@gmail.com

An account with this email already exists

Password: ••••

Repeat Password: ••••



Sign Up

Please fill in this form to create an account.

Name: John Doe

Email: jdoe2@gmail.com

Password: ••••

Repeat Password: ••••

Password is not matching

Code Description

The “SignUp.html” page includes the “Sign.css” stylesheet, which is associated with most account-related forms on the website. It includes“index.js” (which contains common functions) and “AccountData.js” which contains the “createAccount” function.

The “Cancel” button calls “window.history.back()” to return to the previous page, and the ”Sign Up” button triggers the submission event of the form, which calls “createAccount()” and uses the return value to determine whether or not to go to “SignIn.html”

First, the “createAccount” function clears all error messages. It then gets the values of many elements, including the error message elements and the input fields. It first checks if an account using that email already exists, and (if so) shows the error message and exits. Then it checks if the password is repeated correctly, and (if so) creates the account and returns true. Otherwise, it shows the error message and exits.

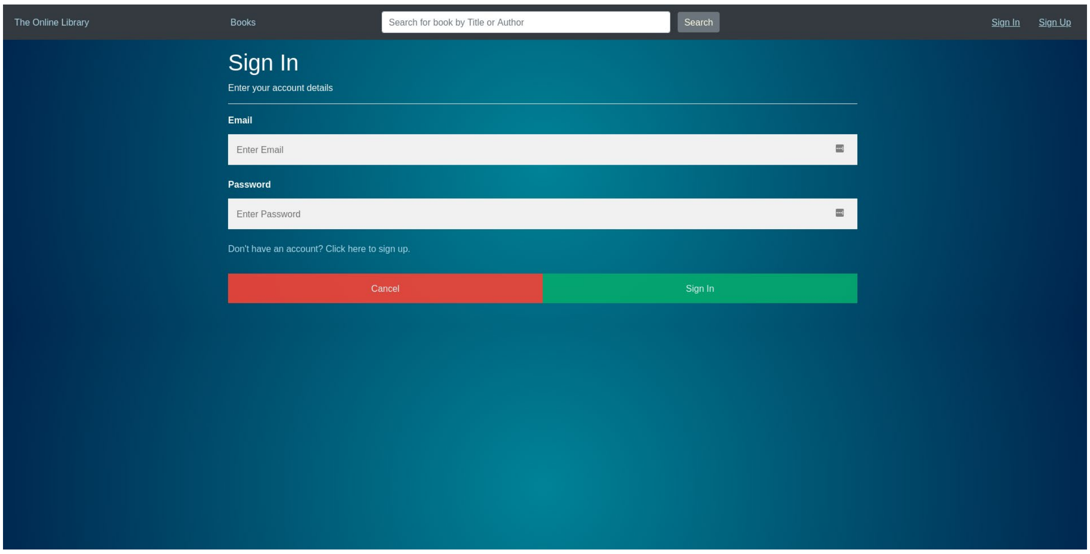
9) Sign In (SignIn.html)

Description

If a user already has an account, they can easily log in by clicking the “Sign In” button on the top-right of the screen. This redirects the user to another page where they can enter their account details. The user can press the “Cancel” button in order to go back to the previous page, if preferred.

If the user tries to log in using an email that is not associated with an account, they get a note stating that this email is not associated with any account. If they fail to enter the correct password, they get another note stating that the password is not correct. However, if all details are correct, the user is logged in and redirected to the homepage.

Screenshots



Sign In

Enter your account details

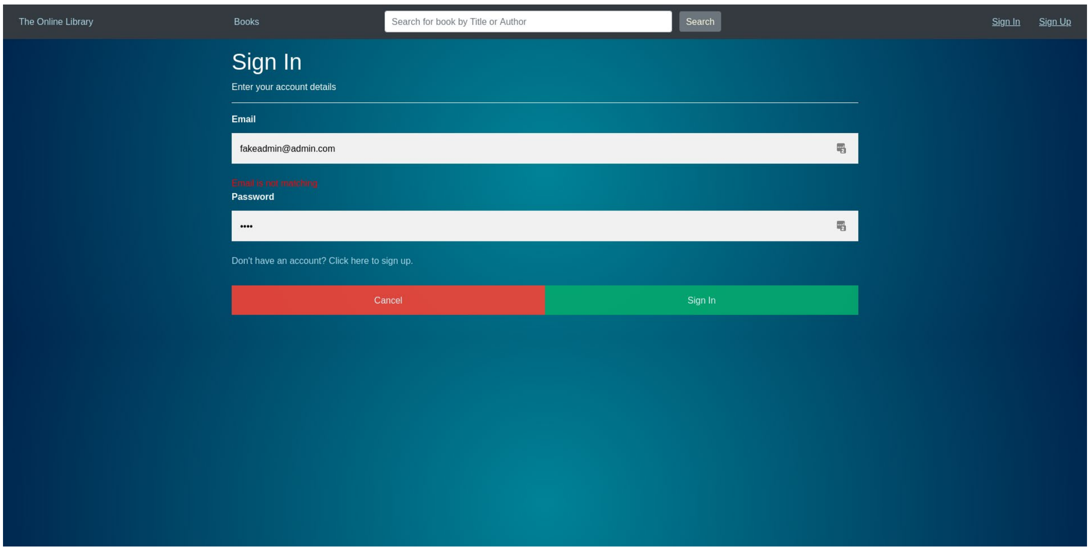
Email

Enter Email

Password

Enter Password

Don't have an account? Click here to sign up.



Sign In

Enter your account details

Email

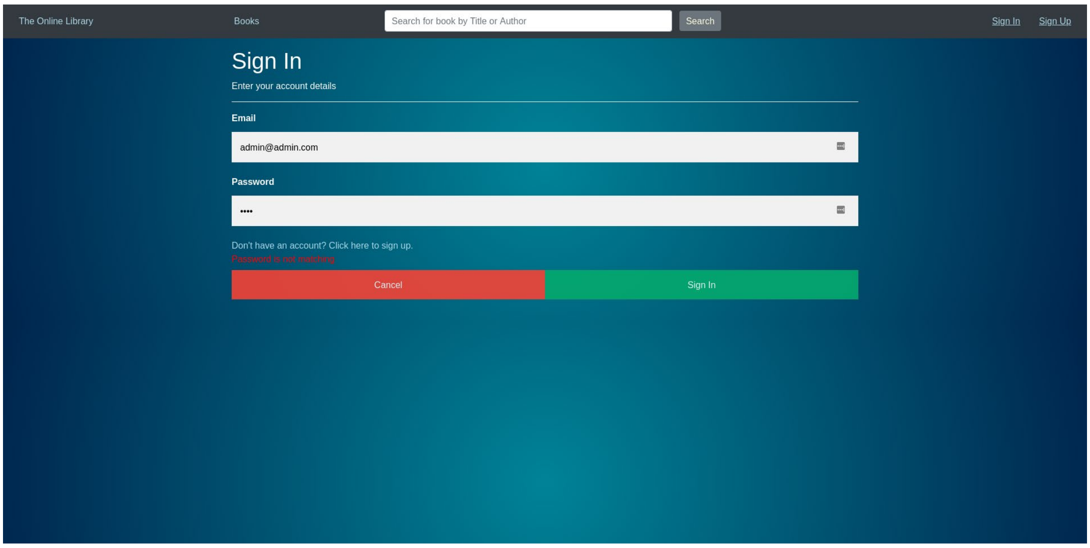
fakeadmin@admin.com

Email is not matching

Password

••••

Don't have an account? Click here to sign up.



Sign In

Enter your account details

Email: admin@admin.com

Password: ••••

Don't have an account? Click here to sign up.

Password is not matching

Code Description

The “SignIn.html” page includes the “Sign.css” stylesheet, which is associated with most account-related forms on the website. It includes“index.js” (which contains common functions) and “AccountData.js” which contains the “login” function.

The “Cancel” button calls “window.history.back()” to return to the previous page, and the ”Sign In” button triggers the submission event of the form, which calls “login(...)” (directly giving it the values of the text fields) and uses the return value to determine whether or not to go to “SignIn.html”

First, the “login” function clears all error messages. It then checks if the email is associated with any account. Once it finds an account with that email address, it checks if the password matches what is given by the user. If no account associated with that email exists, the email error message is displayed. If there is an account associated with that email but the password does not match the stored password, the password error message is displayed. However, if the account is associated with that email and the password matches, the user is logged in and the function returns true (to trigger the redirection to “index.html”).

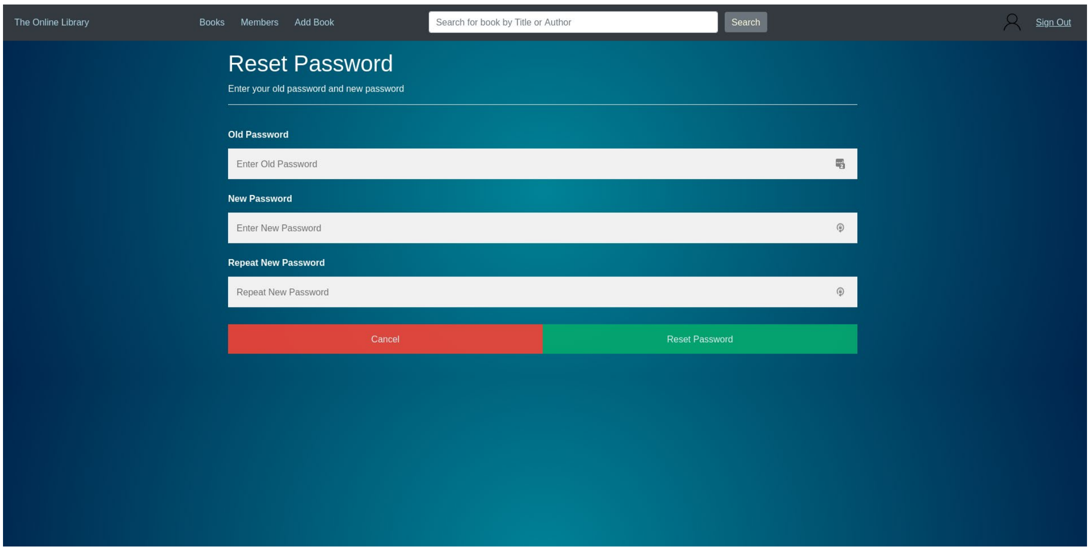
10) Reset Password (ResetPassword.html)

Description

If the user is logged in and wants to change their account's password for security/personal reasons, they can easily access the password reset page by clicking the "Reset Password" on their own profile page. The page requests from the user the old password, the new password, and the confirmation of the new password.

If they fail to enter the correct old password, they get another note stating that the password is not correct. If the user fails to enter the password exactly the same in the "Repeat Password" field, they get another note stating that the password was not repeated correctly. However, if all details are validated, the password is changed, and the user is redirected to the homepage.

Screenshots



Reset Password

Enter your old password and new password

Old Password

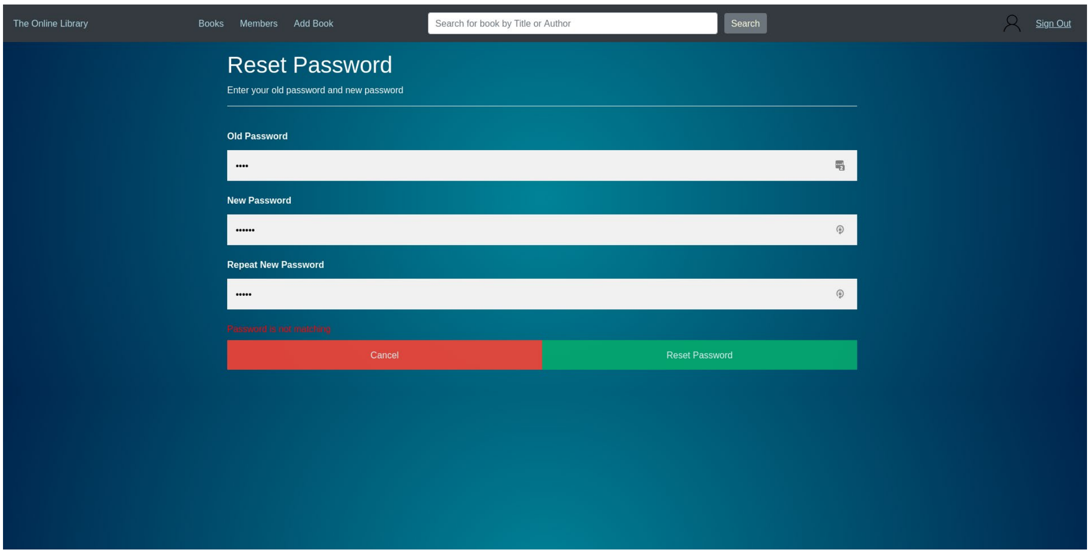
Enter Old Password

New Password

Enter New Password

Repeat New Password

Repeat New Password



Reset Password

Enter your old password and new password

Old Password

••••

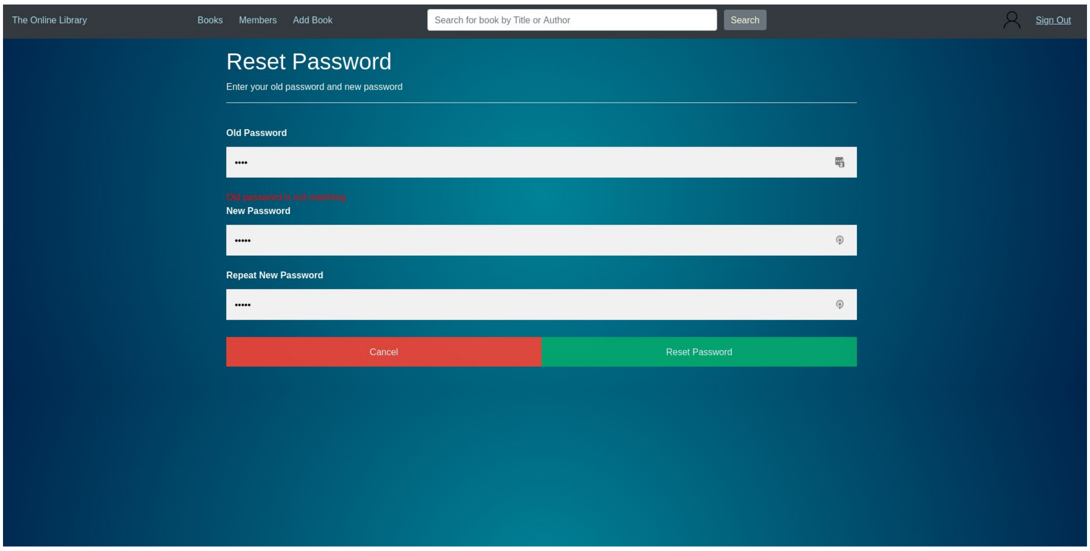
New Password

••••••

Repeat New Password

•••••

Password is not matching



Reset Password

Enter your old password and new password

Old Password

••••

Old password is not matching

New Password

•••••

Repeat New Password

•••••

Code Description

The “ResetPassword.html” page includes the “Sign.css” stylesheet, which is associated with most account-related forms on the website. It includes“index.js” (which contains common functions) and “AccountData.js” which contains the “resetPassword” function.

The “Cancel” button calls “window.history.back()” to return to the previous page, and the ”Reset Password” button triggers the submission event of the form, which calls “resetPassword(...)” (directly giving it the values of the text fields) and uses the return value to determine whether or not to go to “SignIn.html”

First, the “resetPassword” function clears all error messages. It then checks if the user repeated the new password correctly. If not, the function displays an error message and then exits. It then checks if the password of the logged-in account matches what is given by the user. If it does not match, another error message is displayed, and the function exits. However, if the old password is correct and the new password is repeated correctly, the user’s password is changed, and the user is redirected to the homepage.