



---

# INTERNET PROGRAMMING

---



Final Project Report

## *Team Members*

|                         |         |
|-------------------------|---------|
| Mohamed Adel Lotfy      | 18P1724 |
| Adel Asaad Sarofeim     | 18P2949 |
| Abdelrahman Tarek Mahdy | 18P7159 |

## *Chosen Topic*

Online Library

## *Supervisor*

Dr. Hesham Farag

## Contents

|                               |    |
|-------------------------------|----|
| Abstract:.....                | 2  |
| Design & Analysis:.....       | 3  |
| Home Page:.....               | 3  |
| Search button:.....           | 3  |
| Login button:.....            | 3  |
| Logo button:.....             | 3  |
| Profile button:.....          | 3  |
| Members button:.....          | 3  |
| Books button:.....            | 3  |
| Members page: .....           | 4  |
| User/Guest view: .....        | 4  |
| Admin view: .....             | 5  |
| Profile Page: .....           | 6  |
| User view: .....              | 6  |
| Admin view: .....             | 6  |
| Guest (not logged) view:..... | 7  |
| Books Page:.....              | 8  |
| User/Guest view: .....        | 8  |
| Admin view: .....             | 9  |
| Book Details Page: .....      | 10 |
| Search button:.....           | 10 |
| Books button:.....            | 10 |
| Home button:.....             | 10 |
| More from author button:..... | 10 |
| More from Author Page:.....   | 11 |
| Search Page:.....             | 12 |
| Register Page: .....          | 13 |
| Results & Conclusion:.....    | 15 |

## Abstract:

An e-library is supposed to have multiple services such as guest viewership, membership privileges, administrative authority for moderation of both members and books.

This project is a prototype to show the functionalities of an e-library with a simulated database that is reset upon instantiating a new connection to the e-library (database is not saved).

Functionalities included are categorized according to permission of the current user varying from lowest to most permitted through (guest, member, and admin).

Guest functions allow you to view the current members, current list of books, details about each book, more books from the same author.

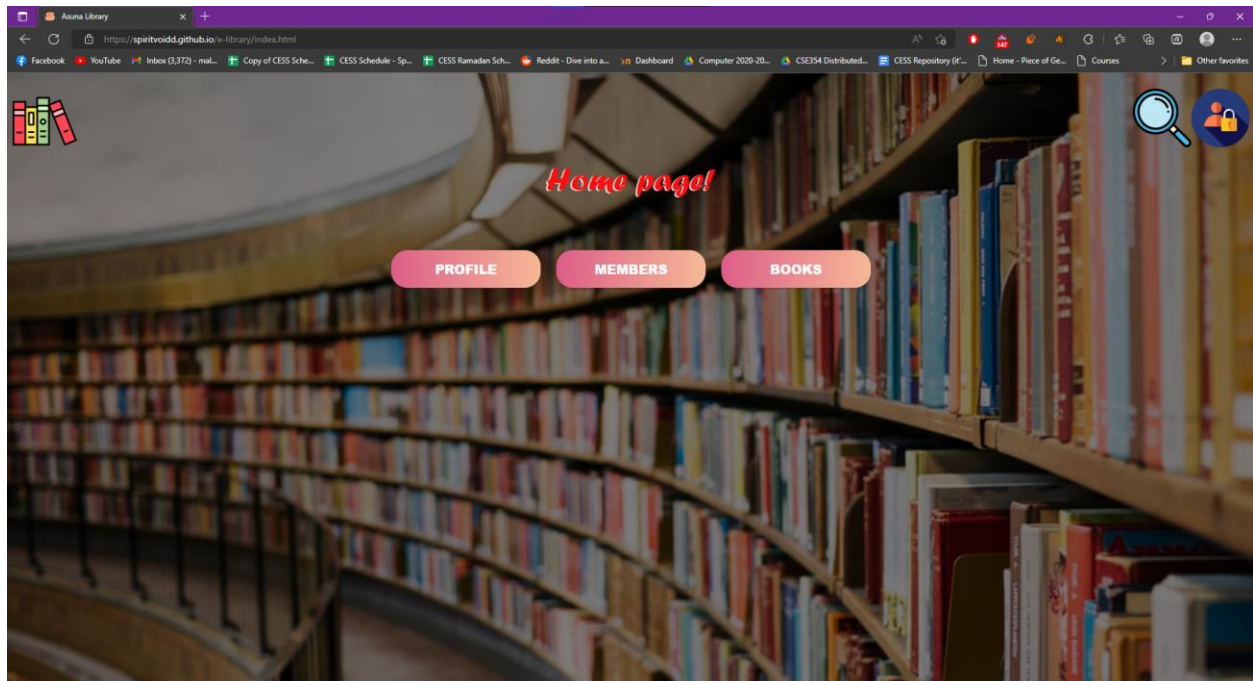
Member functions are the same as guest functions plus you're given a view of your profile in the library.

Administrative functions are the same as member functions plus:

- Moderation over the list of members and books
- Reset button to return the library to a previous saved state

## Design & Analysis:

### Home Page:



Search button:

This button takes the user to the search page.



Login button:

This button takes the user to the login page.



Logo button:

This button exists on all pages to take the user back to the home page.



Profile button:

This button takes the user to the profile page.

Members button:

This button takes the user to the members page.

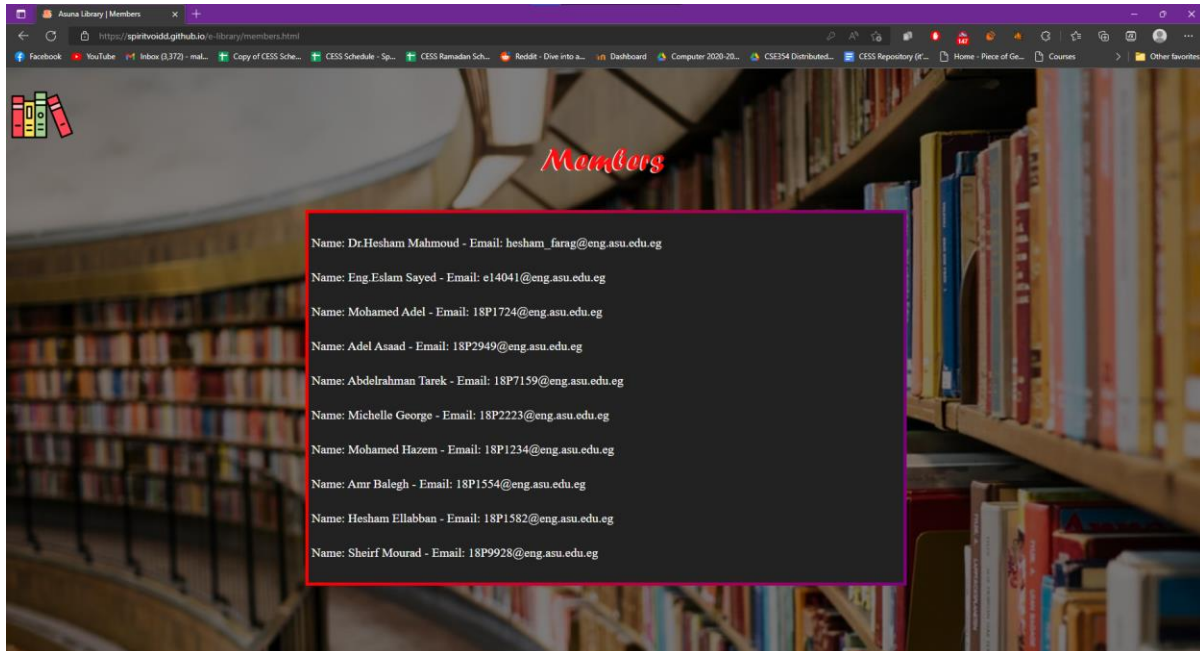
Books button:

This button takes the user to the books page.

## Members page:

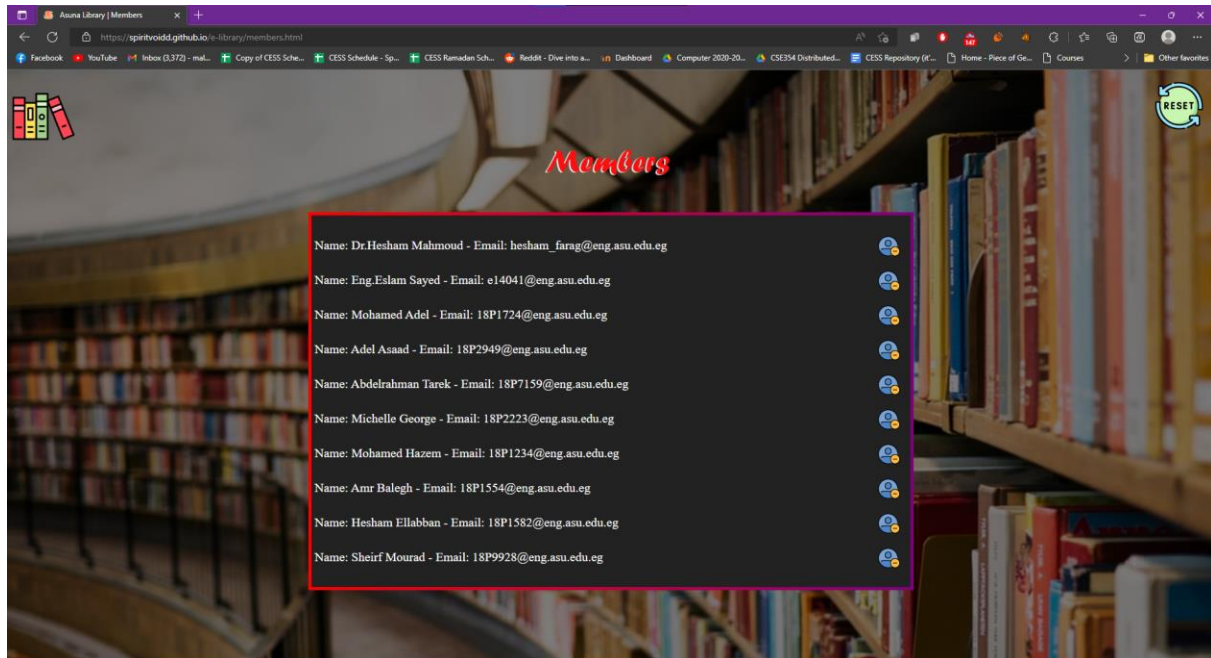
Displays all the users currently in the database.

## User/Guest view:



Users can only see the names and emails of other members.

Admin view:



Admins have the privilege to delete users or reset the users' database to the initial state (the users initial state contains the 10 users displayed in the image above).

*Delete user button:*

This button removes the selected user from the database.



*Reset button:*

This button resets the user's database to its initial state restoring any deleted users.

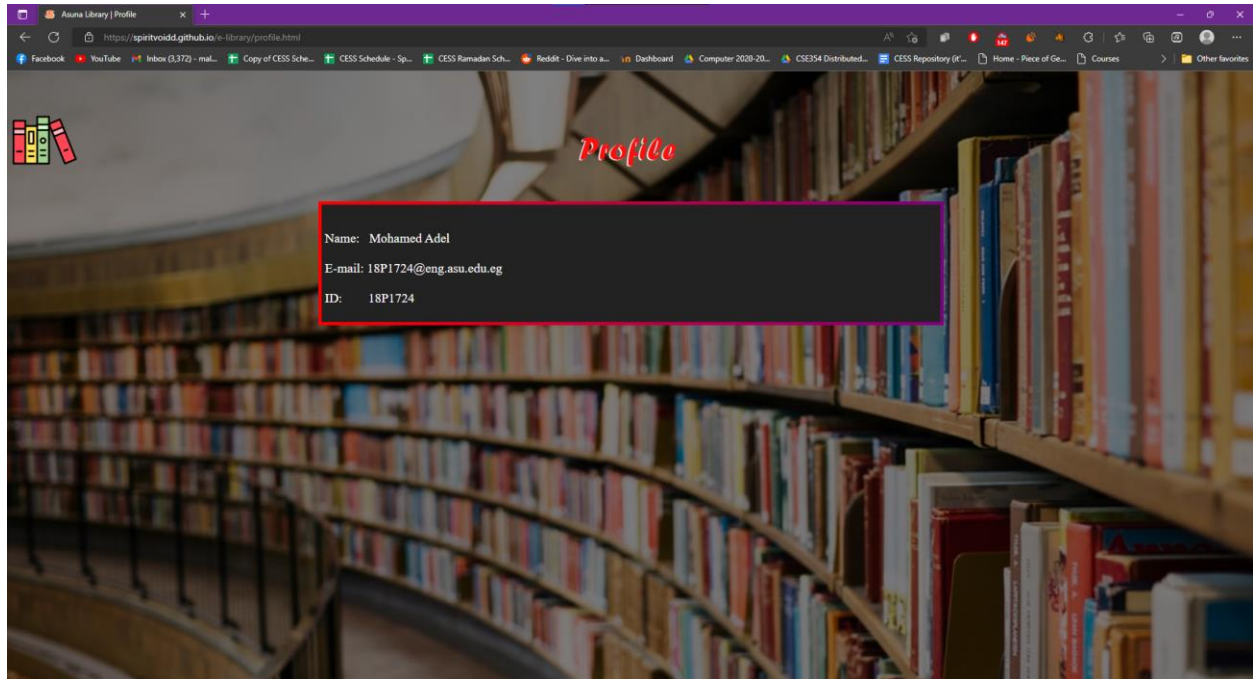




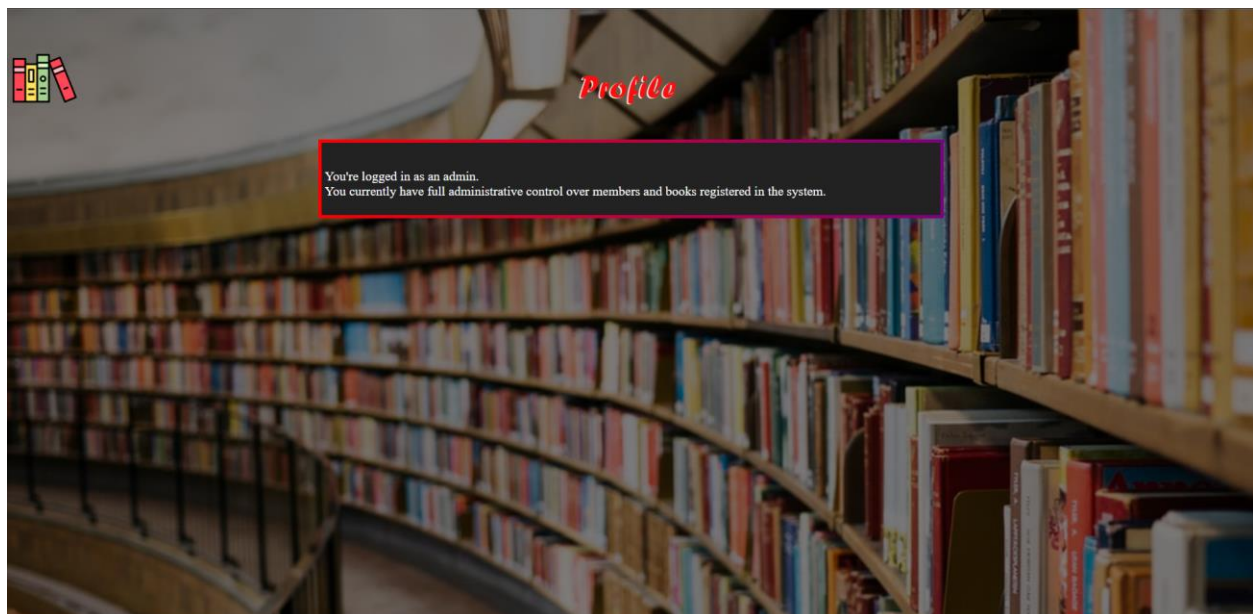
## Profile Page:

Displays the logged in user info.

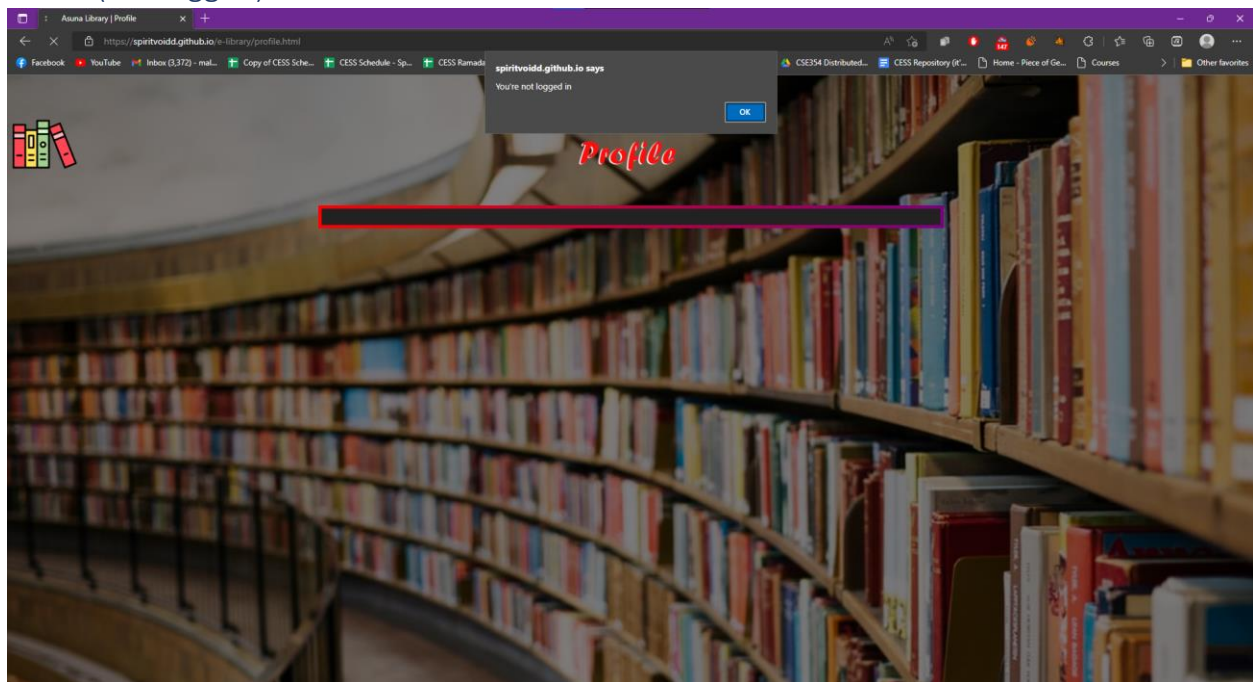
## User view:



## Admin view:



Guest (not logged) view:

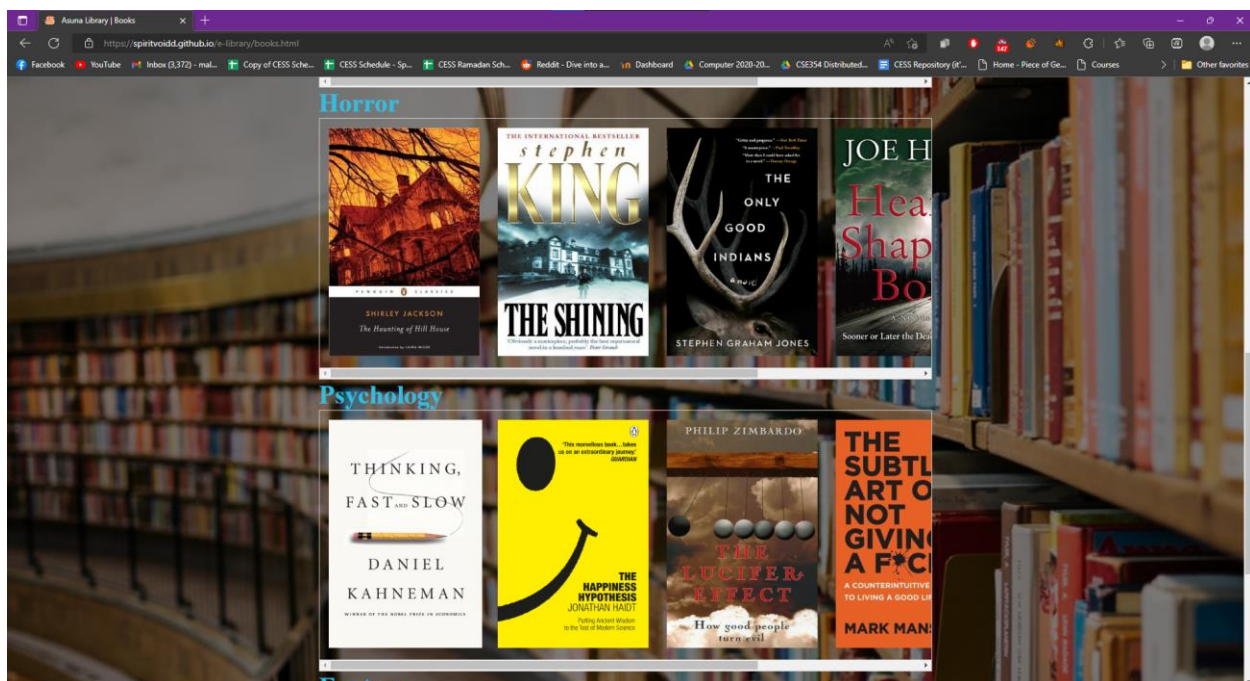
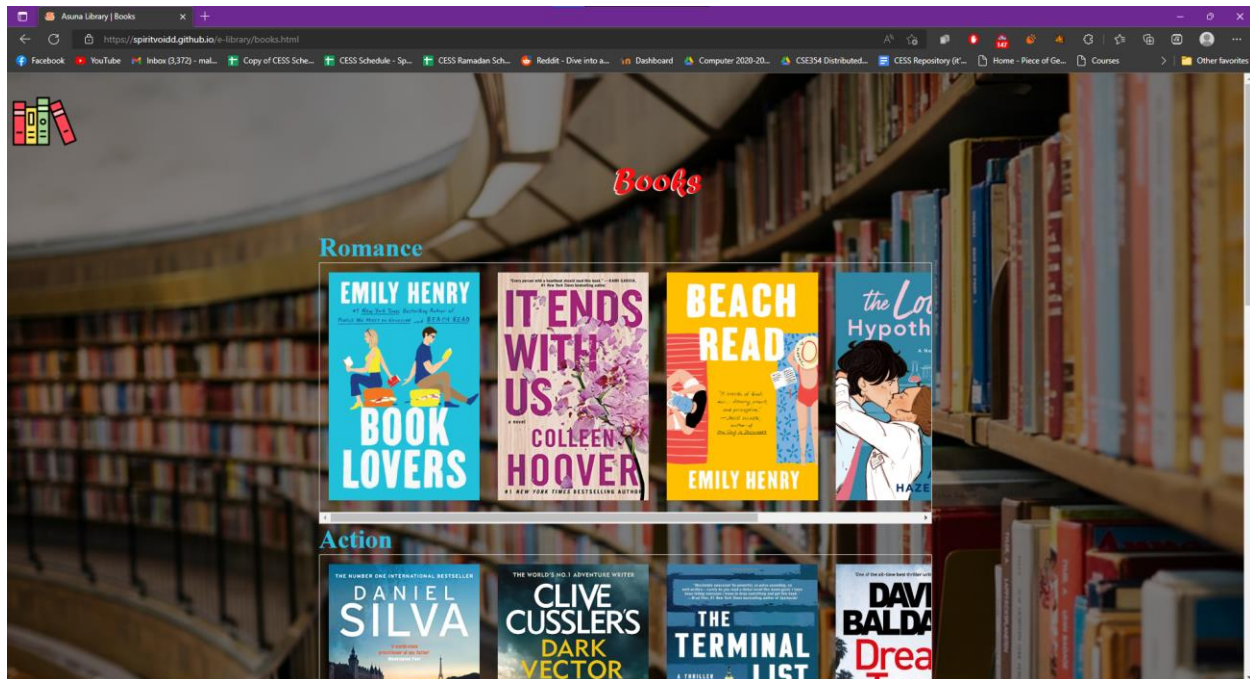




## Books Page:

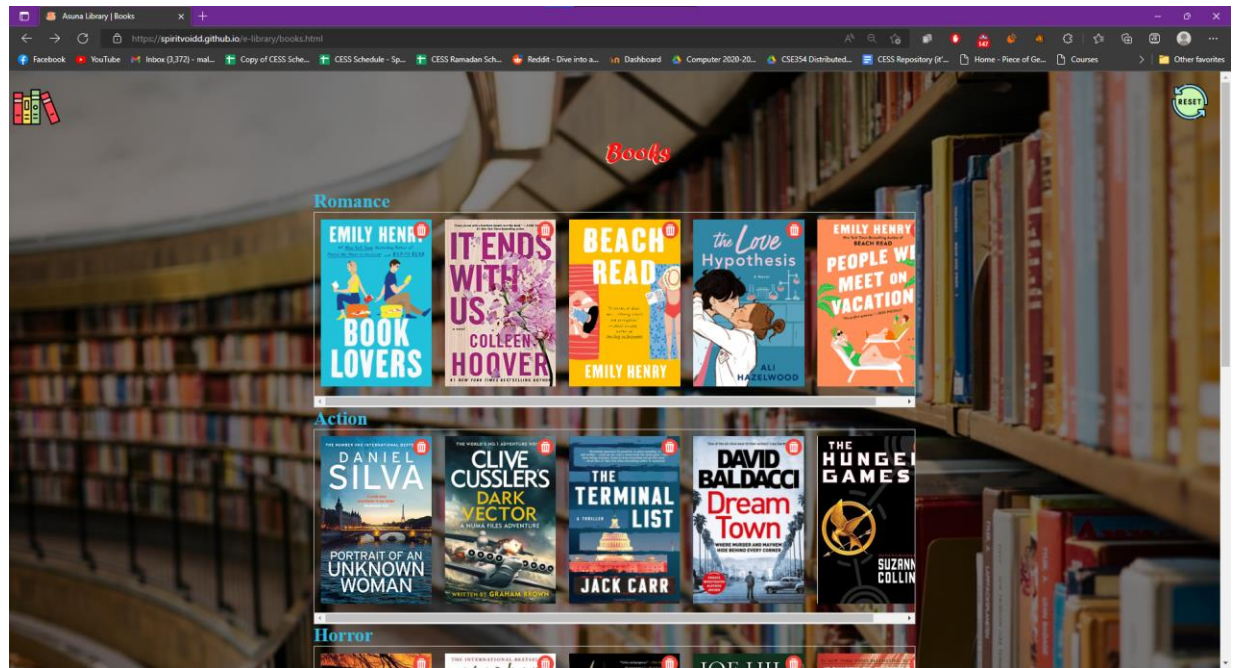
Displays all the books in our database, each book genre is displayed horizontally so the user can scroll horizontally to see all books in that genre.

## User/Guest view:



Admin view:

Admin have the privilege to delete books from the database or reset the books database to its initial state.



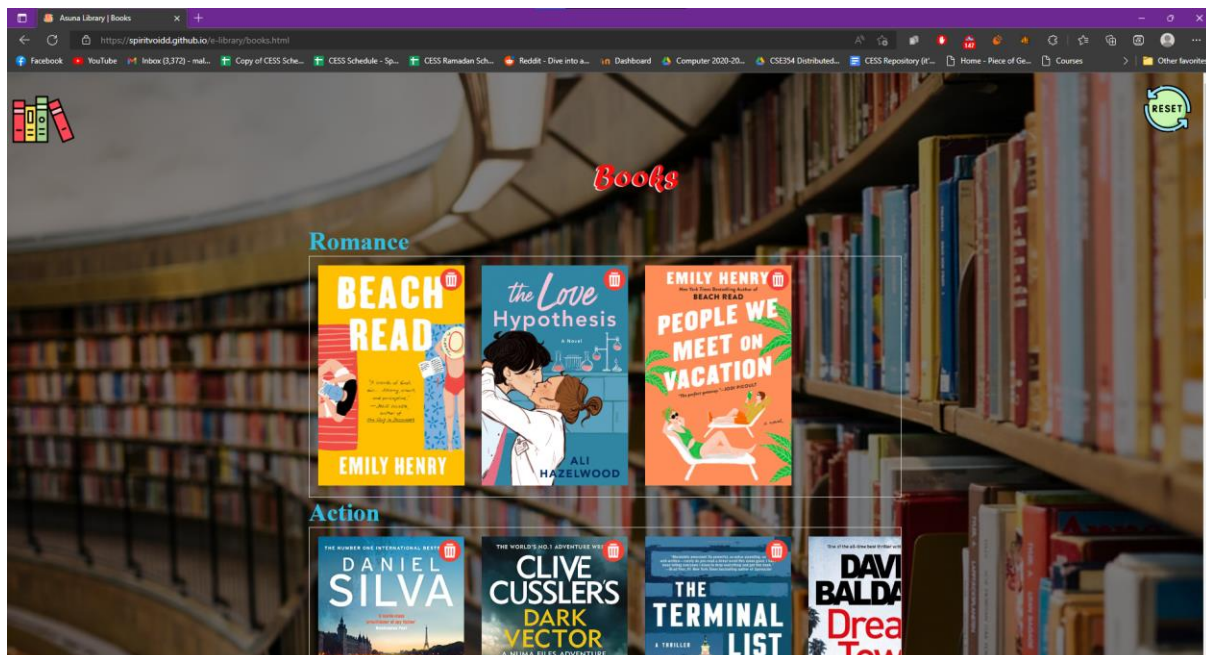
Delete button:

This button deletes the selected book from the database and reloads the page.



Reset button:

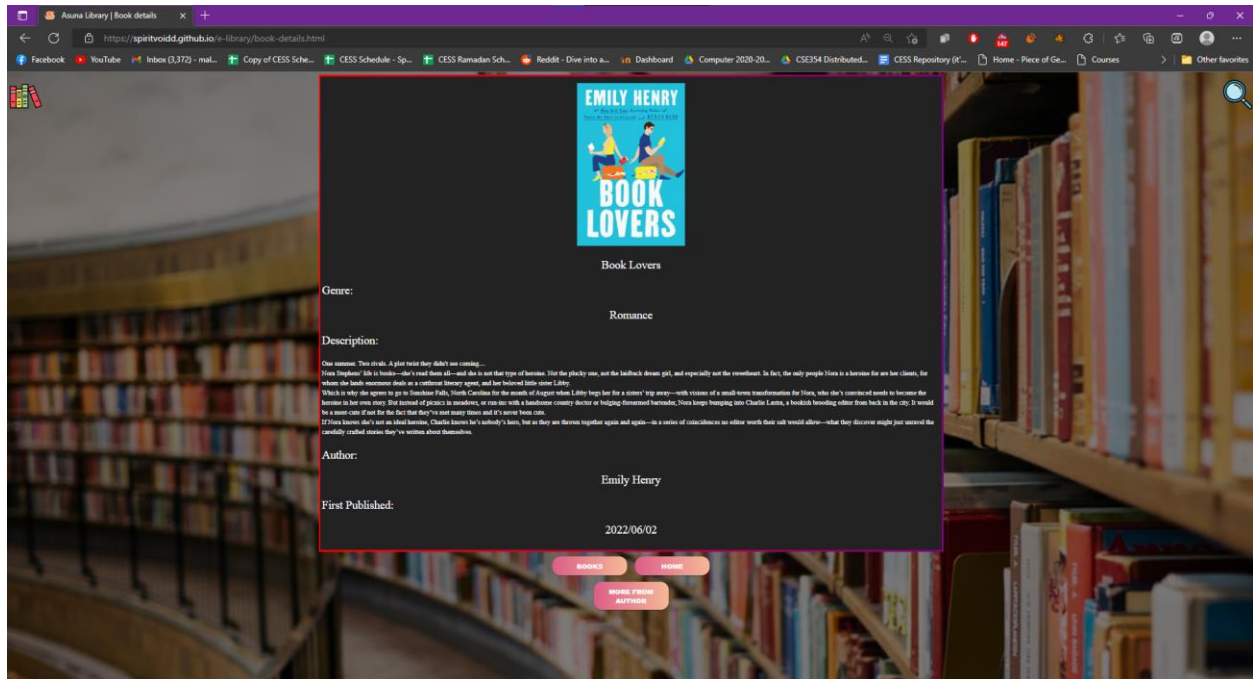
This button resets the books database to its initial state restoring any deleted books.





## Book Details Page:

This page displays all the held information on the book selected by the user.



## Search button:

This button takes the user to the search page.



## Books button:

This button takes the user to the books (previous) page.

## Home button:

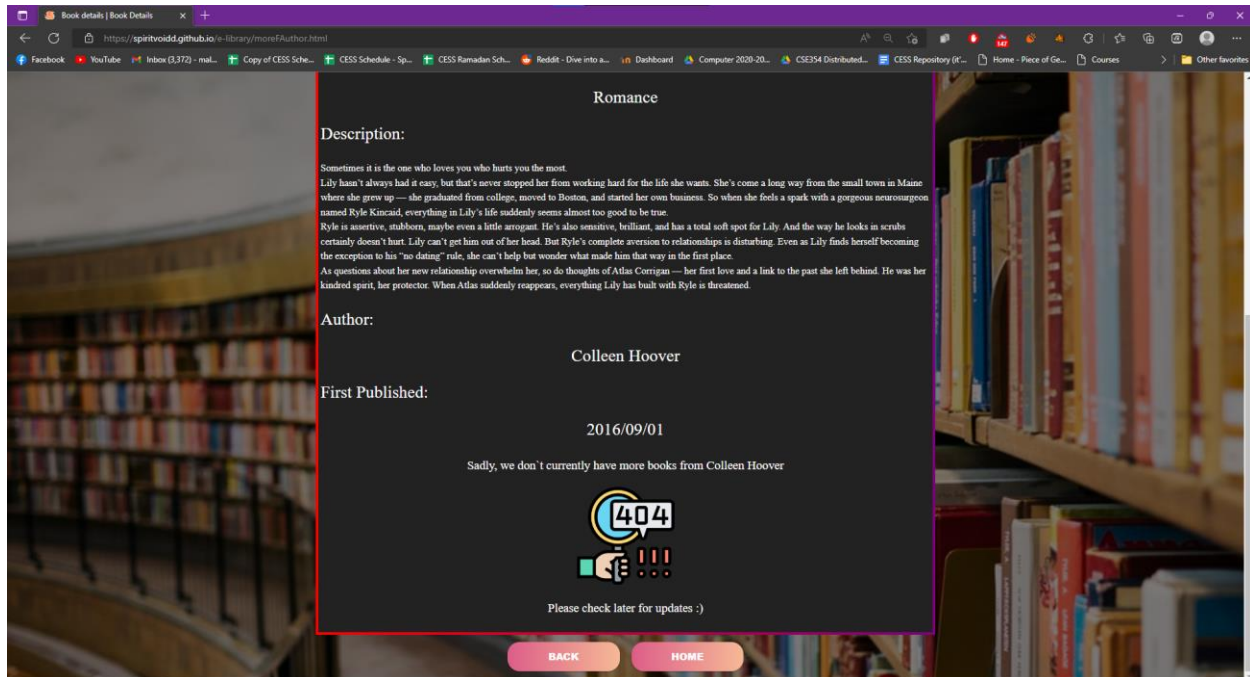
This button takes the user to the home page (it does the same function as the logo on the top left corner, but we needed it because book details when zoomed in the user may have to scroll down a lot so instead of scrolling upward just to go home now, he has a button to make it quicker).

## More from author button:

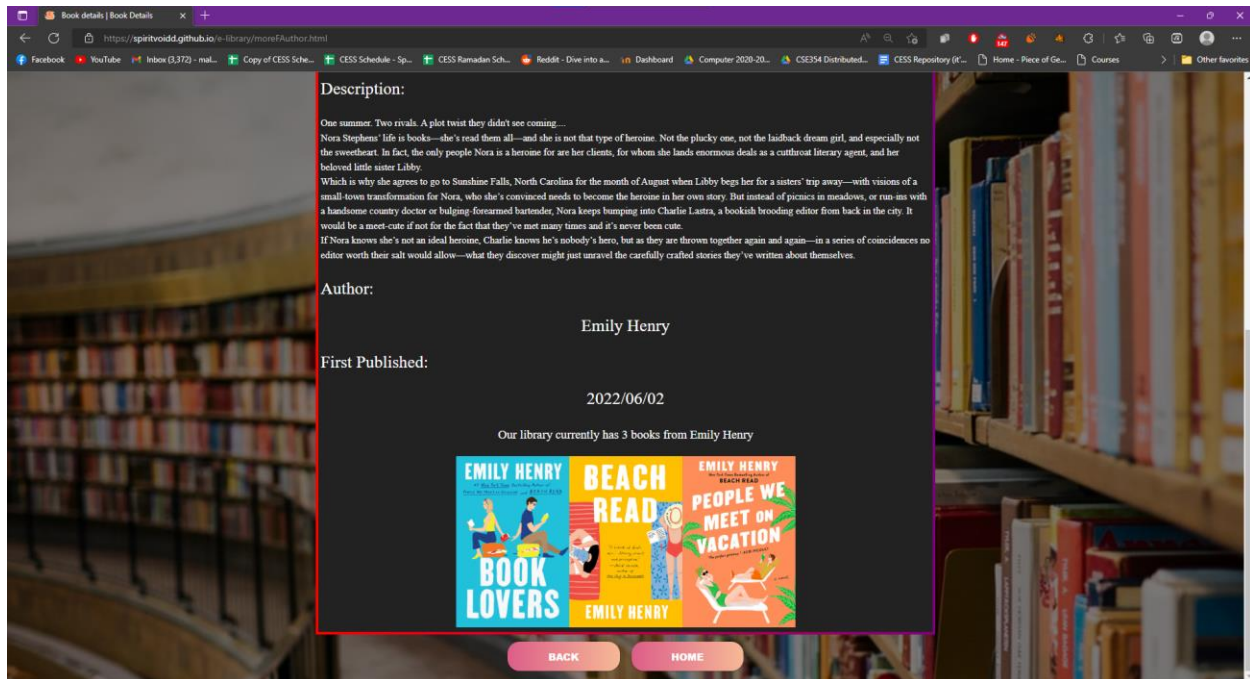
This button displays the number of books for this author in our database and it even prints the image of these books (making them clickable to see their details as well).

## More from Author Page:

Displays error message if the author has only one book in the database.

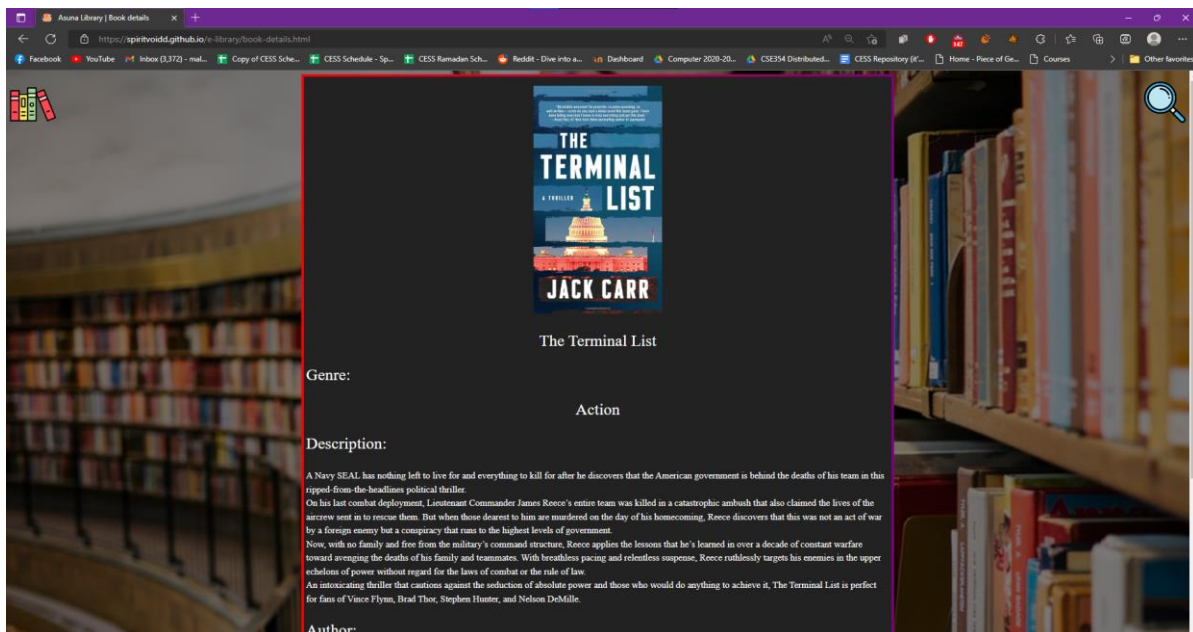
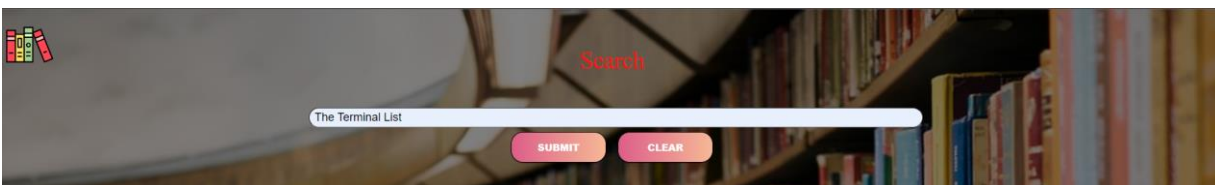
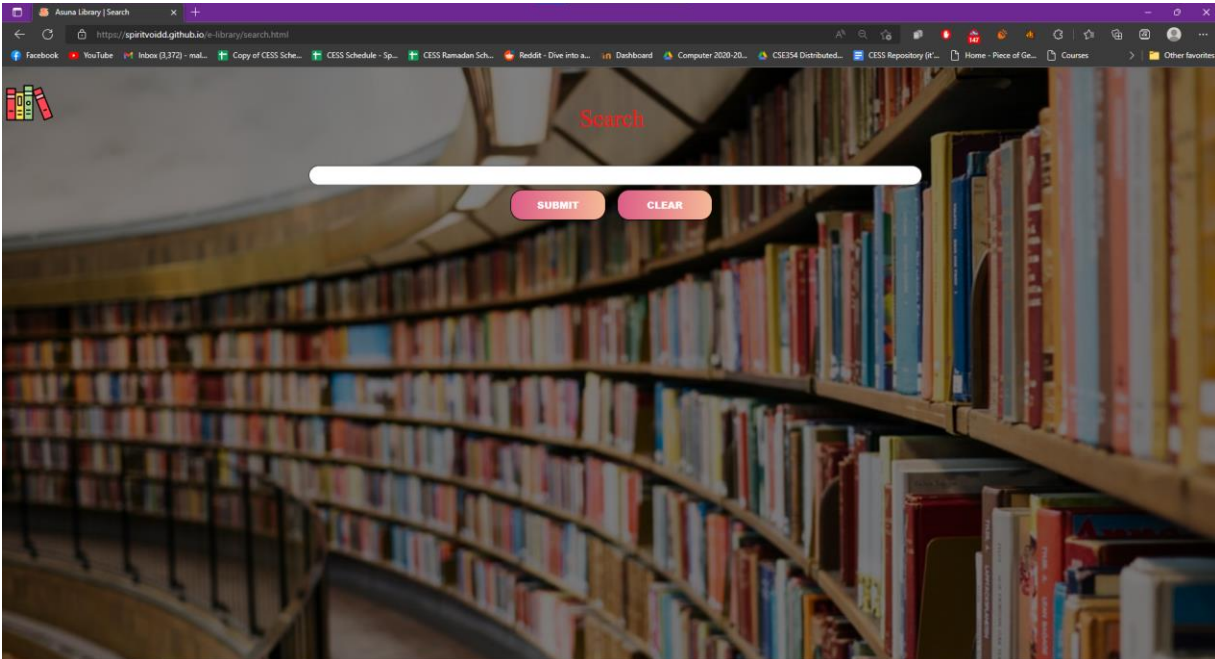


Displays all author's books in our database.



## Search Page:

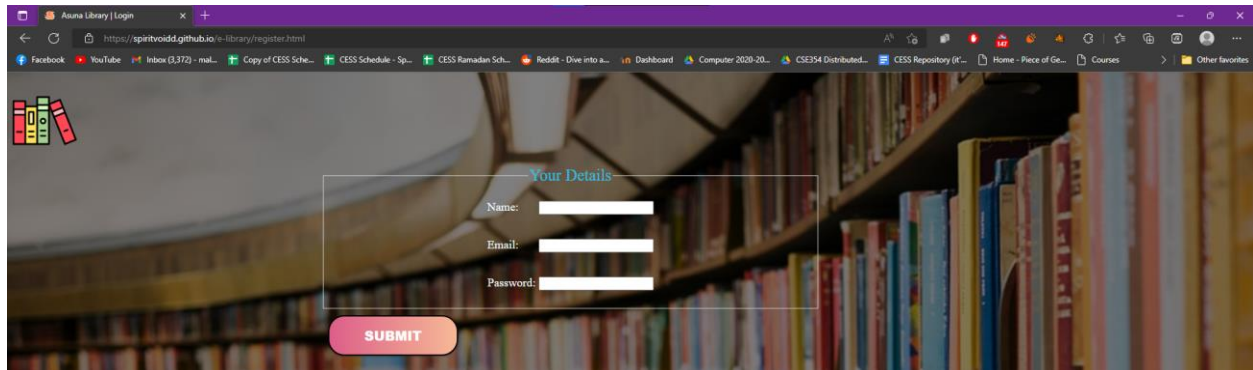
This page searches the books database for the name written by the user, if found then that book's details page is displayed.





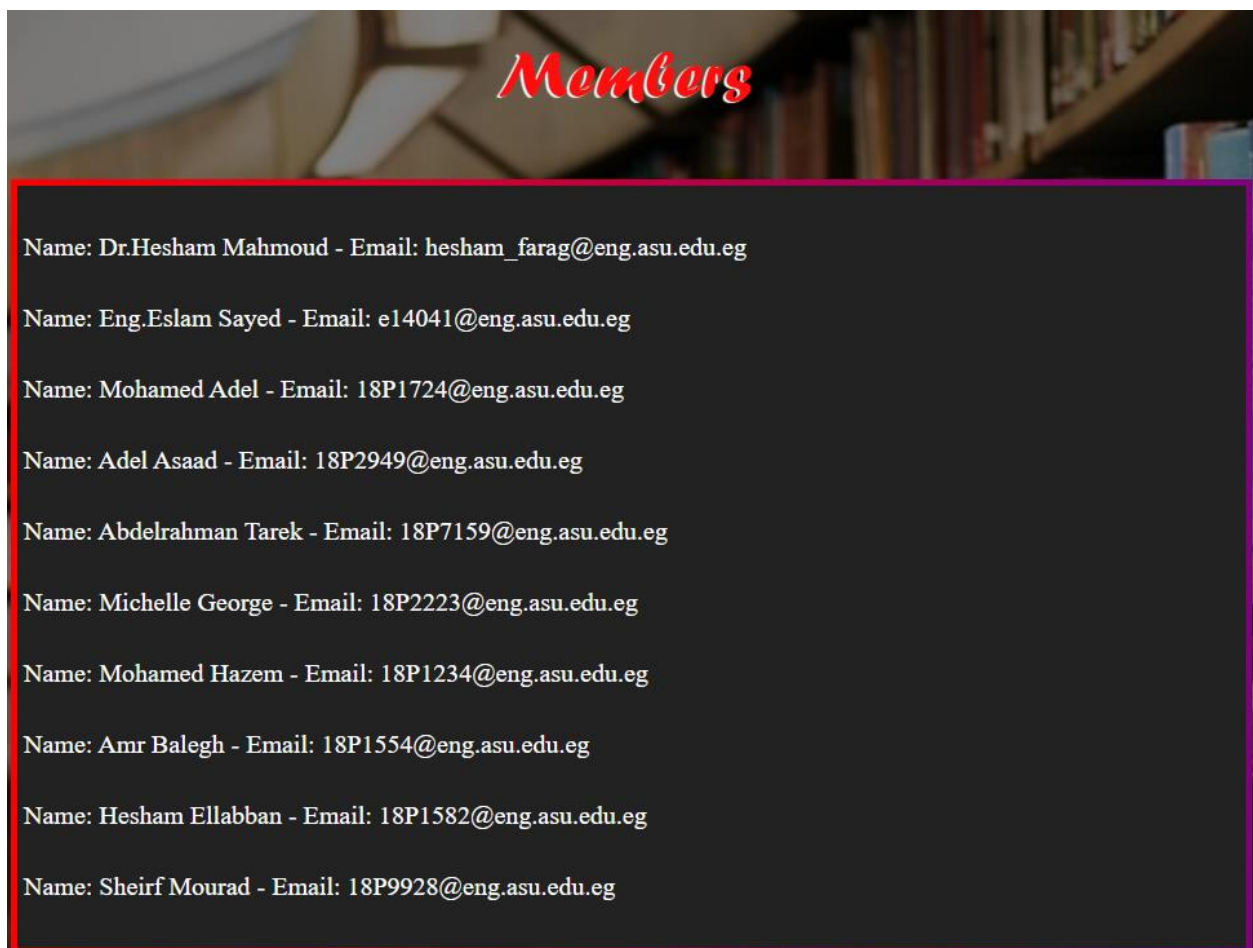
## Register Page:

Takes input from the user and adds a new user to the user's database.



A screenshot of a web browser showing the 'Asuna Library | Login' page. The browser's address bar displays 'https://spirtvoidd.github.io/e-library/register.html'. The page features a background image of a library with bookshelves. In the center, there is a registration form titled 'Your Details' with three input fields: 'Name:', 'Email:', and 'Password:'. Below these fields is a pink 'SUBMIT' button. The browser's tab and address bar are visible at the top, and various icons are present in the top right corner.

Screenshot of members page before registering.







**Your Details**


Name:

Email:

Password:

**SUBMIT**

Screenshot of members page after registering.



**Members**

Name: Dr.Hesham Mahmoud - Email: hesham\_farag@eng.asu.edu.eg

Name: Eng.Eslam Sayed - Email: e14041@eng.asu.edu.eg

Name: Mohamed Adel - Email: 18P1724@eng.asu.edu.eg

Name: Adel Asaad - Email: 18P2949@eng.asu.edu.eg

Name: Abdelrahman Tarek - Email: 18P7159@eng.asu.edu.eg

Name: Michelle George - Email: 18P2223@eng.asu.edu.eg

Name: Mohamed Hazem - Email: 18P1234@eng.asu.edu.eg

Name: Amr Balegh - Email: 18P1554@eng.asu.edu.eg

Name: Hesham Ellabban - Email: 18P1582@eng.asu.edu.eg

Name: Sheirf Mourad - Email: 18P9928@eng.asu.edu.eg

Name: register test name - Email: register@yahoo.com

## Results & Conclusion:

Different programming techniques were used to simulate a database without actually using a database.

One of which was the usage of *sessionStorage* function that is included from the Window object, we preferred *sessionStorage* over *localStorage* to store the simulated database so as to not use any more space.

Of course, in real-life situations neither of those would be used for a database, instead SQL and PHP would be used to store the database safely, which is why this is a simulated database, not a real one.

*sessionStorage* had a lot of keys for different values including (permission key, books in the library key, members registered key) which were the three mainly used database keys in the project.

CSS was used to specify what to do when hovering over certain objects as they would light up or show differently, and the cursor pointer would change accordingly to make the GUI more intuitive and user-friendly.

The pages were almost all generated dynamically save for a few structural elements that were unnecessary to add in the dynamic generation of pages as they were always there no matter the user input.

You can switch between **guest** view, **member** view, and **administrative** view (and edit) through the login function provided in the project.

- For **guest** view: logout (default)
  - For **member** view: login using an e-mail and password of one of the members, or register a new one (hint: we put the password as the ID of the member for ease-of-access)
  - For **admin** view: login with the e-mail as "admin" and the password as "admin"
- Note: We set the admin view this way for ease-of-access, in reality this would be much more complicated so as to not make a security breach.

## Project link:

Project is uploaded to Github and you can access the functionalities of the project through [here](#).

You can also view the source code of the project through [here](#).