



WHERE LITERATURE COMES TO LIFE, TOGETHER

Submission 2

by

Rose Luo s2549840

Eoin Reid s1858933

Vaishnavi Srinivasan s2601954

[Screencast Link - UniRead](#)

# IntRODUCTION

UniRead isn't just about reading - it's a community where book lovers come together to share their love for literature. Everyone can log in, read a book, and share their thoughts.

What makes UniRead special is how the community chooses which book to read each week. Instead of using algorithms, everyone works together to pick the next book. Whether it's talking about conspiracy theories, analyzing plots, sharing favourite lines, or discussing the parts they didn't like, members dive deep into the books they find on UniRead.



Our idea for UniRead came from wanting to bring people together as books brought us together. We talked about books and realized we wanted a place where people could discuss them and read them too. What started as a simple quotes page became a whole website with a library, voting system, and discussion page. We wanted a place where users could share their favourite quotes, help pick which books to read together, and talk about them. By working on UniRead together, we hoped to show our strengths, learn from each other, and create something meaningful.



The methodology in developing UniRead involved a collaborative effort achieving both good design and technical functionality. Rose and Vaishnavi headed the design aspect, incorporating feedback from the first submission. They completed most of the website's design and models before the first submission, refining them based on suggestions.

They worked majorly on the user experience. Additionally, they enhanced the landing page to provide comprehensive information about the website and maintained design choices by modifying titles accordingly. They also re-worked on creating consistency in design between the different sections of the website.

Eoin managed the technical side, creating a voting system using AJAX. He also made a space for reading the weekly book and integrated a like and dislike system on the discussion page to encourage engagement.

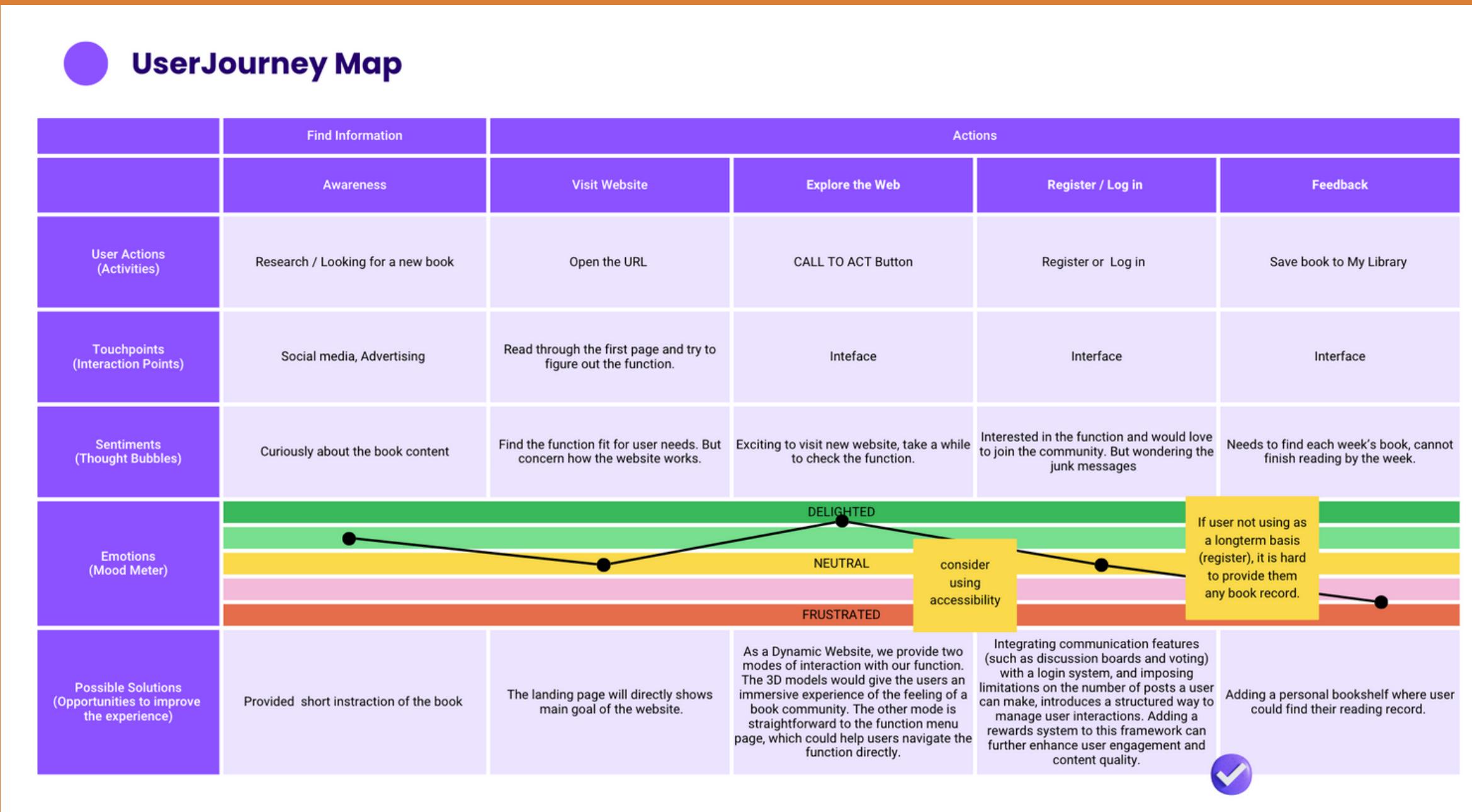


# DESIGN STRATEGIES

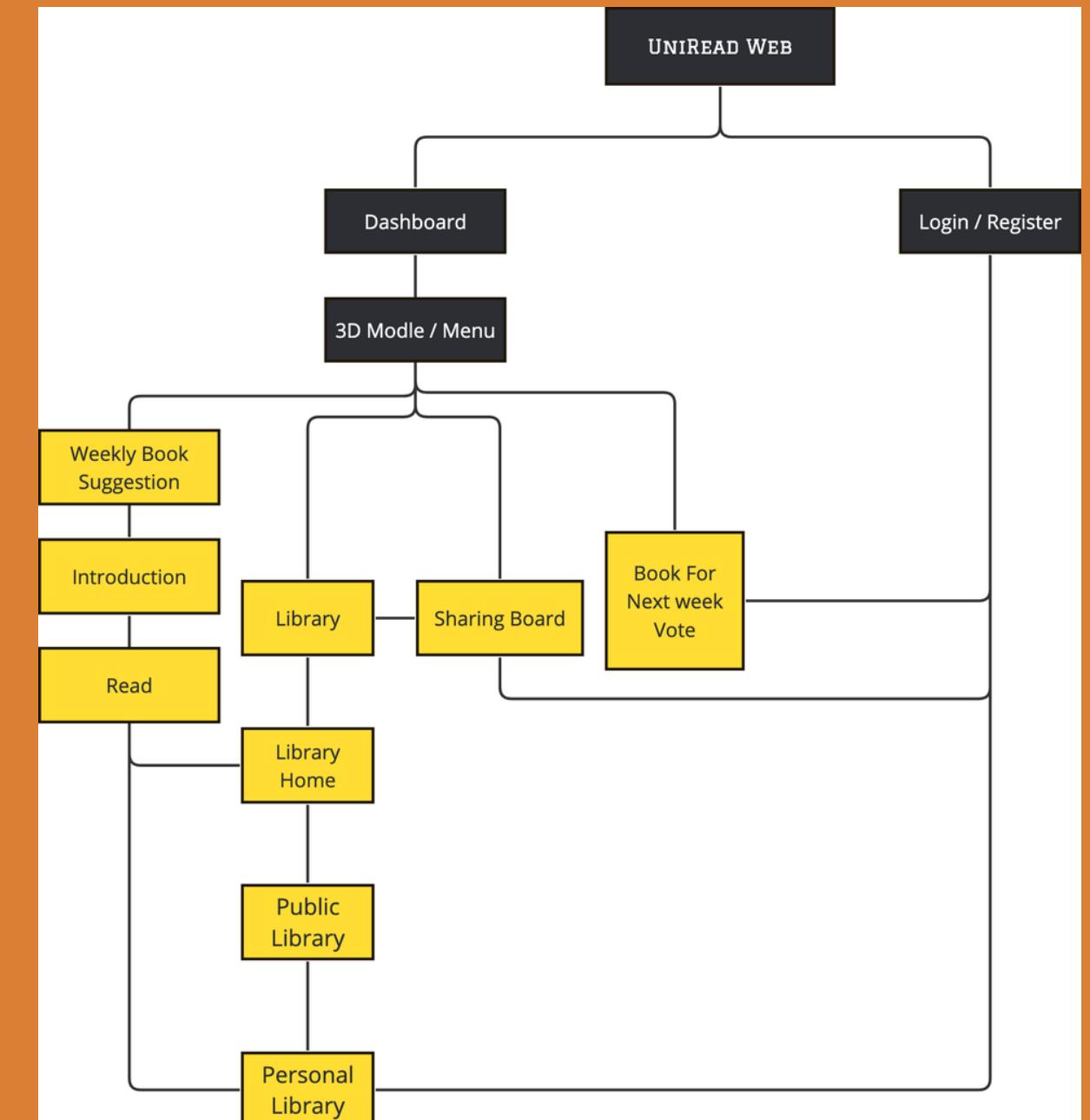
- General functions (Features): Generating the primary function and secondary function which could solve the user's frustration. Always provide the bake buttons to allowing users the freedom to explore, make choices in the website.
- Technology Constraints: Ensure the application is accessible to individuals with disabilities.
- Cross Channels: Content strategy, working along with different media. such as 3D modeling and sound effect. Thinking forward, we also could link with Shopping website, witch could helps user to find different version of the book.

# User Experience

## User Journey Map



## Site Map





The first stage in implementing our new functionality was to design the relevant database structure so, the following tables were created-

- google\_2 containing a list of around 300 books storing ID, Title, ISBN, and haveRead, a binary marker indicating if the book has been read (collected from <https://1001bookreviews.com/the-1001-book-list/> and nextBooks - [https://www.kaggle.com/datasets/bilalyussef/google-books-dataset?resource=download&select=google\\_books\\_dataset.csv](https://www.kaggle.com/datasets/bilalyussef/google-books-dataset?resource=download&select=google_books_dataset.csv)), storing three books that are pulled from google\_2 being voted on, storing title, ISBN, voteCount and URL
- This week's book, pulled from the winner of last week's nextBooks table (stores only the ISBN)
- Personal library storing ID, ISBN, username, and a binary flag for whether the book is favourited.

- A user table, to store usernames and hashed passwords for security
- A discussion table, to store comments users make about books.

Once the database structure was ready, we first created the login and register structure, this allows users to interact with all the new functionality and to prevent users from manipulating votes by repeatedly voting. Checks now happen before voting, visiting the personal/favourite libraries, and leaving a comment. When a user creates an account, their details are stored in the database, and their password is hashed using the build in password\_hash(\$data["password"], PASSWORD\_DEFAULT); password hashing functionality. This ensures that we offer protection for the passwords that we store. When users log in, their password is hashed and checked against the database and on success, a session is created with a user variable that's passed to other pages.

# Different Pages and their Functionalities

## This Weeks Book "*thisweeksbook.html*":

The page connects to and queries the 'this week's book' table in the database and collects the relevant ISBN for this week, this ISBN is then used to query the Google Books API, collecting details about the book. These details are then added dynamically to the page through a Java and php script updating the page in front of the user. The user also has the option to view a book reader through another connection to the Google API and has the option to view this on a full-screen page if the text is too small.

## Discussion Board "*discuss.html*":

This page makes a similar query to the database retrieving the ISBN and connecting to the Google Book API to dynamically display this week's book details. We then have a form in which users who are logged in can leave a comment, when they do so the book is automatically added to their personal library (the only way of doing so is to encourage engagement). At the bottom, we can see all the community's comments in a list where users can like/dislike comments updating the database.

## Different Pages and their Functionalities

**Library "publicLibrary.html, privateLibrary.html, favourites.html":**

To implement the library we found a style of bookshelf which was exactly the look we wanted at <https://tympanus.net/Development/3DBookShowcase/index2.html>, so a lot of the styling and CSS was adapted from this, however, they fill out the books with placeholder information, and our goal was to dynamically fill the books from the database and google books API so this presented a big challenge. This was probably the most time-consuming aspect of the code update but the final result is effective. For the public library, the google\_2 DB is queried to return only ISBNs for books where haveRead is 1, we then pass this ISBN list to an event listener which connects to the Google API and fills information into a nested structure for the books which are dynamically added to the page. This process is the same for the personal library and favourites except they query API endpoints that we made which return books the current session user has read and/or favourited.

Lastly, the style for many of the pages was updated along with suggestions made in tutorials and CW1 feedback such as a landing page which appears only if the user isn't logged in, loading indicators, fixes to the burger menu, larger icons and buttons, added alt text to all key images for usability e.t.c.

## Further Development and Project Critique

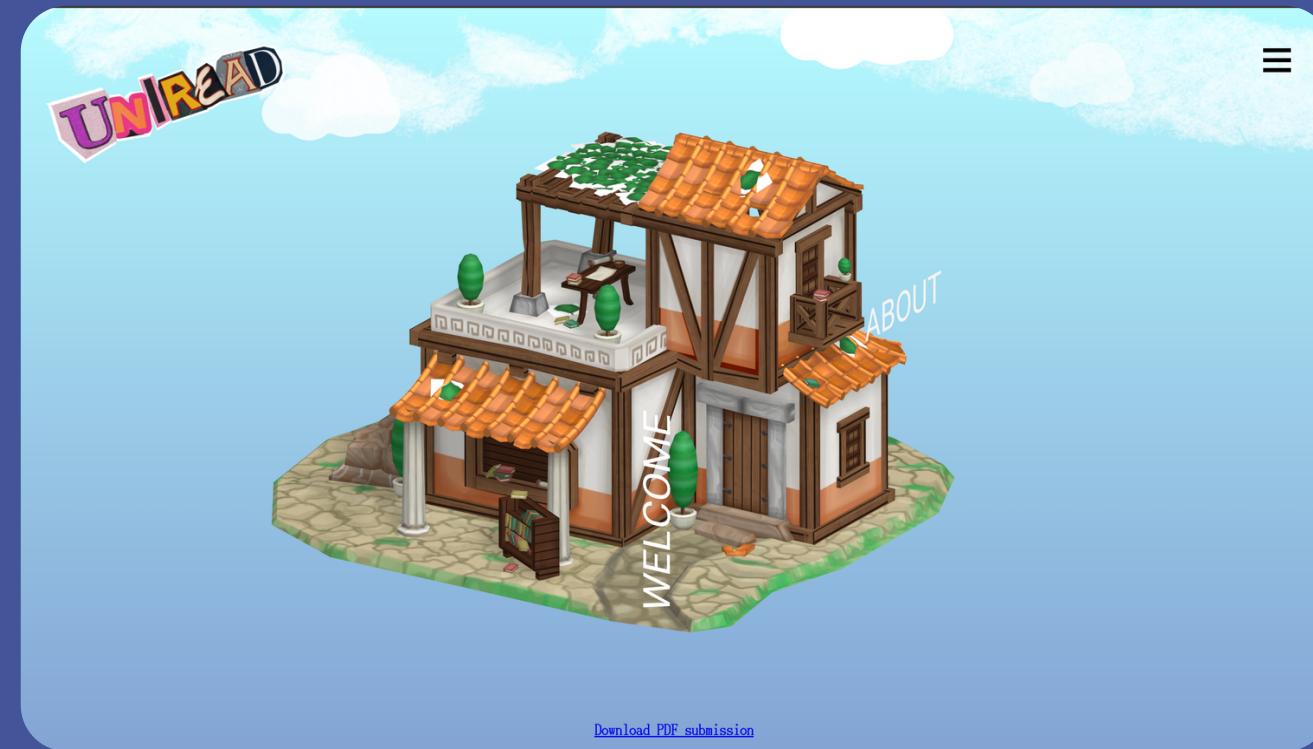
There is a clear recognition within our team that there is significant potential for user engagement and improving the overall user experience on UniRead. Specifically, the current voting page is relatively basic and can be further developed to provide users with more interactive voting experiences.

We also see the opportunity to enhance the discussion page by implementing threaded conversations, allowing users to reply to specific quotes or comments. Looking ahead, we envision the possibility of a personal chat feature, enabling book readers to communicate directly with each other. We also didn't manage to optimise image file size, although we removed some big images.

Furthermore, we recognise the potential benefits of hosting our library of books instead of depending on 3rd party sources. With this approach, we could implement features such as linking directly to specific lines or pages within books, thereby making conversations more specific and enriching the overall user experience.

OUR WEBSITE

# Home



# Library Home



Log In



Register



Home

≡ Library

- Library Home

- Public Library

- Personal Library

- Favourites

This Weeks book

Discuss

Vote

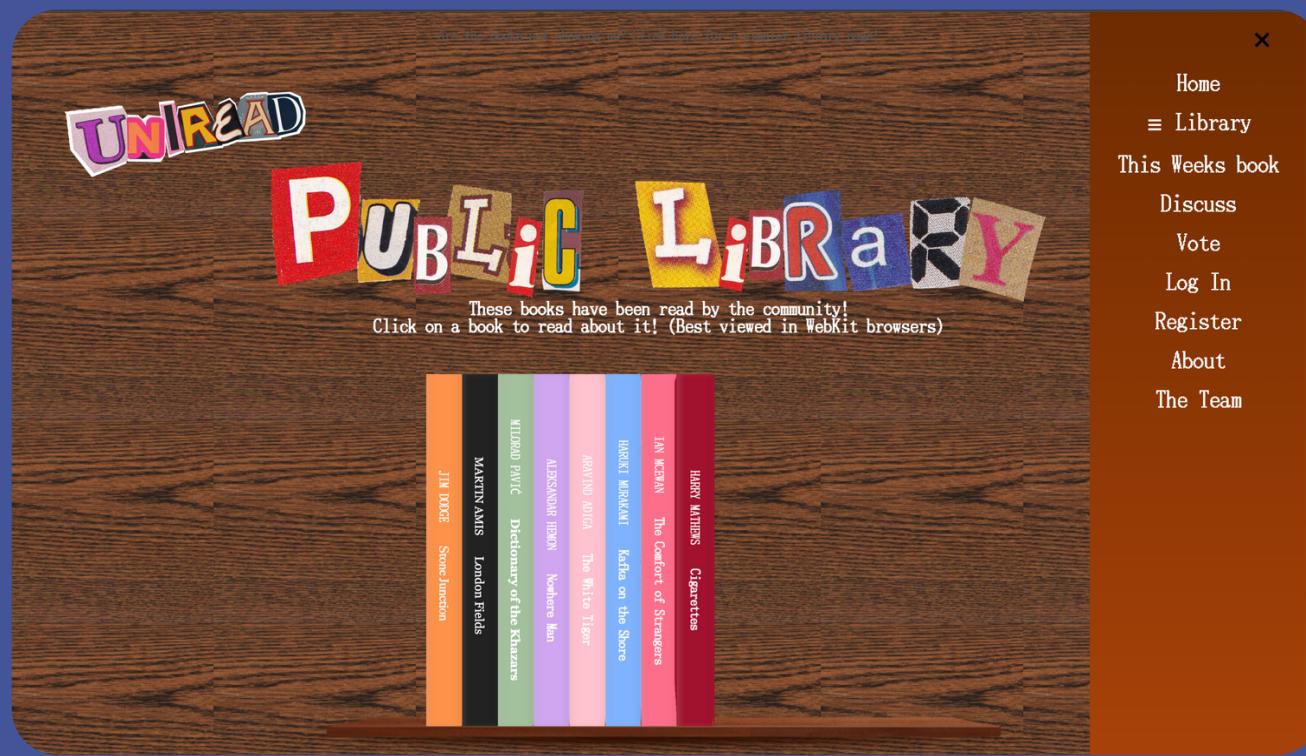
Log In

Register

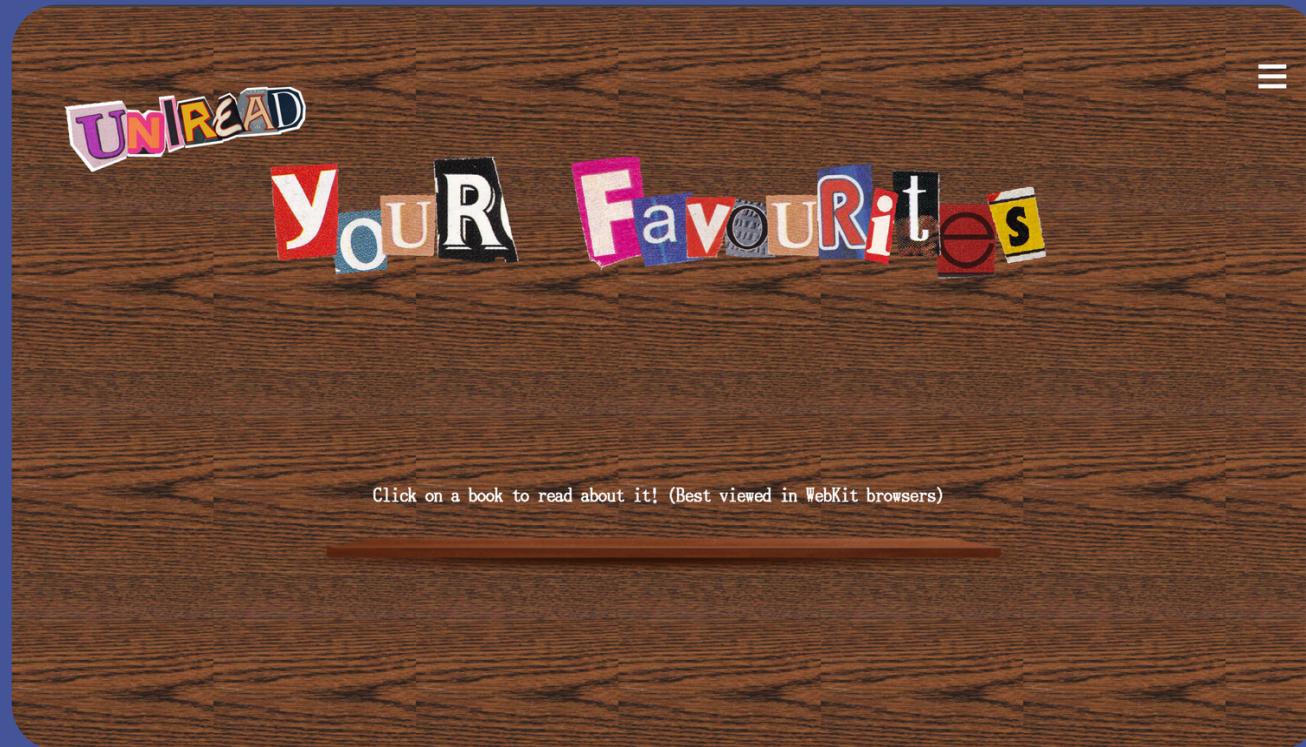
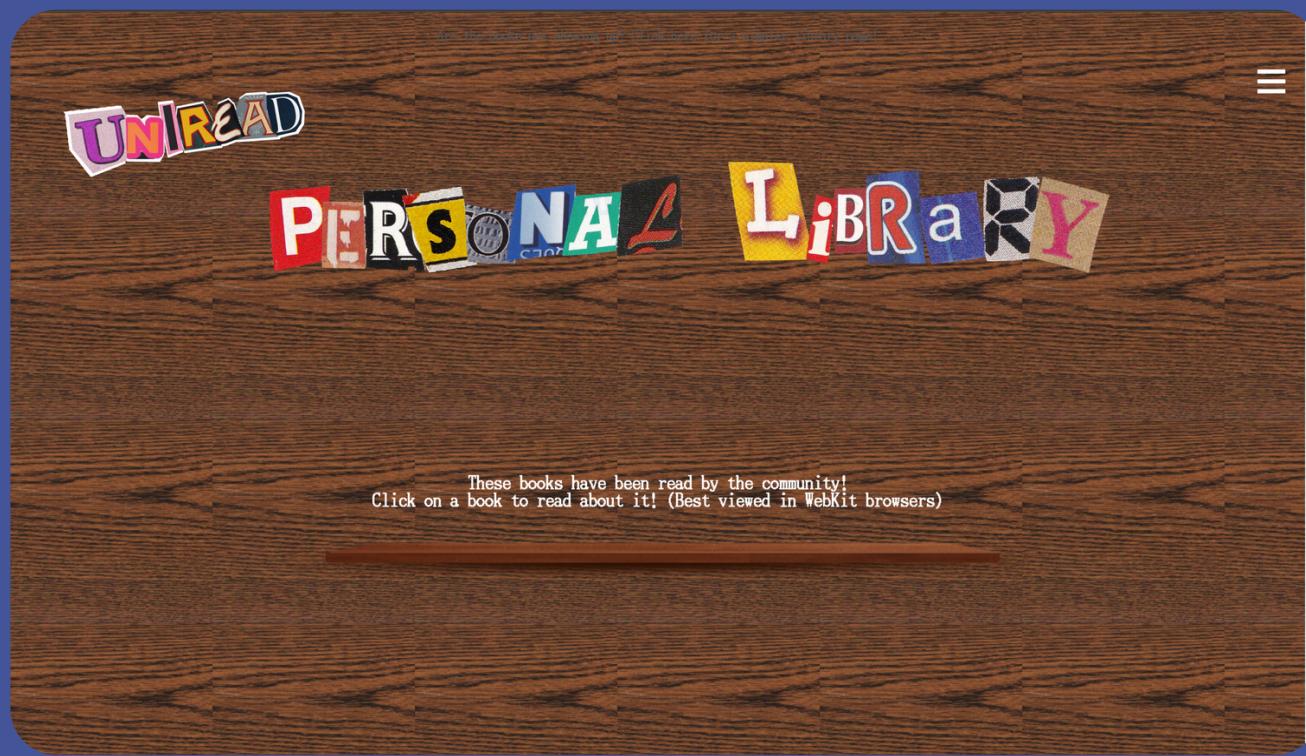
About

The Team

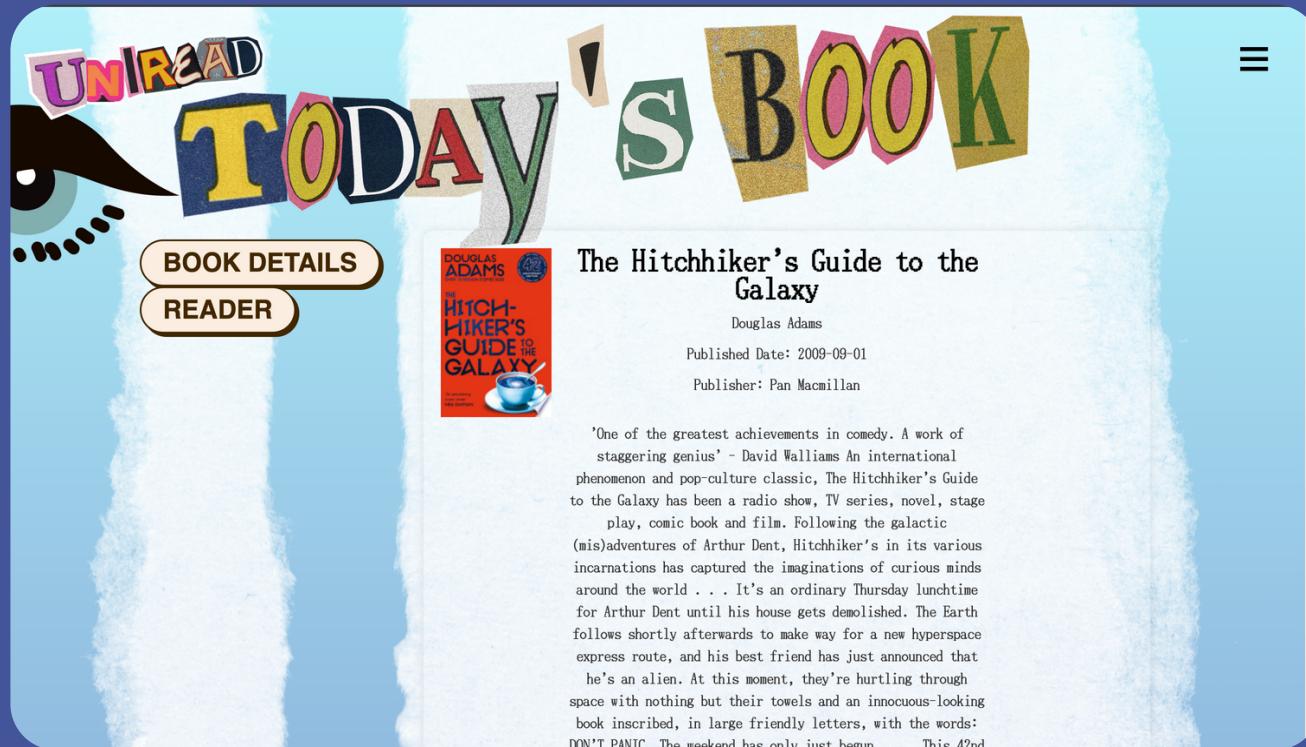
# Public Library



# Personal Library



Favourites



Today's Book



Home

≡ Library

- Library Home

- Public Library

- Personal Library

- Favourites

This Weeks book

Discuss

Vote

Log In

Register

About

The Team

# DISCUSS

This Weeks Book:

The Hitchhiker's Guide to the Galaxy by Douglas Adams

What did you think of this weeks book?

# DISCUSS

What did you think of this weeks book?

By leaving a review you automatically add this book to your personal library!

Your thoughts...

Send Reset

Community Reviews:

Sort reviews: Likes (Descending)

Sophie

"I loved this book it's timeless classic exploring the decadence of the Jazz Age. Fitzgerald's language paints a vivid, unforgettable picture of 1920s America"

Like Count: 42

Tony

VOTE

Vote for next weeks book:

Cover Photo	Title	Votes	Action
	Crime and Punishment	64	<a href="#">Upvote</a>
	Alice's Adventures In Wonderland	61	<a href="#">Upvote</a>
	The Brothers Karamazov	58	<a href="#">Upvote</a>

Welcome to uniREAD

This is a dynamic online Book Club, for book lovers to read, share, discuss, and explore literature together! Let's start!

Click and drag on the book club building to interact and follow the links!

[log in or register.](#)

Vote

Landing Page

Home

≡ Library

- Library Home
- Public Library
- Personal Library
- Favourites

This Weeks book

Discuss

Vote

Log In

Register

About

The Team

# REFErENCES

Kanther, J. Book Preloader, source code, Available from:  
<https://codepen.io/jkantner/pen/mdMydQO>. [Accessed 5 Apr 2024].

Aerne, D. Css Vars Styling, source code, Available from: <https://codepen.io/meodai/pen/rNedxBa>.  
[Accessed 5 Apr 2024].

Anon, 2012. The 1001 Book List. 1001 Books to Read Before You Die [online]. Available from:  
<https://1001bookreviews.com/the-1001-book-list/> [Accessed 5 Apr 2024].

Llic, M 2015. 3d book showcase , source code and images. Codrops [online]. Available from:  
<https://tympanus.net/codrops/2013/01/08/3d-book-showcase/> [Accessed 5 Apr 2024].

Anon, 2024. Google Books Dataset [online]. Available from:  
<https://www.kaggle.com/datasets/bilalyussef/google-books-dataset> [Accessed 5 Apr 2024].

Anon, 2024. Accessibility: Image Alt text best practices [online]. Siteimprove. Available from:  
<https://help.siteimprove.com/support/solutions/articles/80000863904-accessibility-image-alt-text-best-practices> [Accessed 5 Apr 2024].



# REFERENCES

Wood Texture, Media Materials, TextureTown - Download Textures! [online]. Available from: <http://textures.neocities.org/> [Accessed 5 Apr 2024].

Title Letters, zolee, 2022. Magazine Letter Cutouts Ransom Note (PNG Transparent). *OnlyGFX.com* [online]. Available from: <https://www.onlygfx.com/magazine-letter-cutouts-ransom-note-png-transparent/> [Accessed 5 Apr 2024].

Escudero, A. Clouds Moving Css Animation, source code, <https://codepen.io/antonioescudero/pen/zrxGve>, [Accessed 5 Apr 2024]

Pixabay (2024). PageTurn. [mp3.] Available at: <https://pixabay.com/sound-effects/search/flip/> [Accessed 4 Apr. 2024].

Font used throughout site- Ano 2024. homebrew-cask-fonts/Casks/font-437-plus.rb at master, GitHub. Available from: <https://github.com/Homebrew/homebrew-cask-fonts/blob/master/Casks/font-437-plus.rb> [Accessed 5 Apr 2024].

Debugging, Anon., 2024. ChatGPT [online]. Available from: <https://chat.openai.com> [Accessed 5 Apr 2024].

# REFERENCES

An example of the chatgpt usage would be :

Showing it my html source and get/post function for the home page and asking "How do these interact? Is there a possible reason the html page isn't loading despite the code looking right"

## and output

Potential issues in your code include:

### 1. Incorrect Conditional Logic in F3 Route Handler

- In the PHP snippet, the condition to check the session for a user is present, but regardless of the outcome, `$f3->set('loggedIn', false);` is set, which might not be what you intend. This logic essentially tells the application that the user is never logged in, which could be part of why the login modal is always displayed.

### 2. Incorrect Variable Use for Login Status

- The HTML attribute `data-logged-in="{{ @loggedIn }}` seems intended to reflect the user's login status, but the provided server-side script sets this variable to false unconditionally. Make sure that the server-side