

CS 623 Cloud Computing Final Project Report

Book Review Website

Project by: Saida Afroj (NetID: mc3299)

[Project on cloud Link](#)
[GitHub code repository link](#)
[Presentation Video](#)

Background	2
Problem	2
Solution	2
How to Run	2
Website	2
Running Locally	3
Project Architecture	3
FrontEnd	3
Database	5
Design	5
Why Relational database instead of NoSQL	5
Cloud Configuration	6
Implementation & Features	6
User Interface and User experience	6
Organized by categories	7
Adjustable UI for screen sizes	7
Responsive Book Display Grid	9
Advanced CSS	9
Smooth Review Page	10
Search & Recommendation	11
Advanced Search	11
User Based Recommendation	14
Trends	14
Current Cloud Version Limitation	14
Future Improvements	14

Friend List with Activity Logs	14
Machine Learning Models	15
Automated Data Collection	15
Important Links	15

Background

This report is about the Project *Book Review Website*, which was created for graduate course CS 623 *Cloud Computing* in Fall 2021 session at CSUEB. This project end-to-end is done by [Saida Afroj](#) , NetID: mc3299.

Problem

With the internet is becoming the world's largest communication centre, it has provided an environment where everyone can be online at any given time. Online book review sites can be useful for book lovers to buy the kind of books they are interested in . A growing body of research indicates that reading literally changes your mind. This habit offers so many benefits including building vocabulary, increasing ability to empathize, reduce stress, alleviate depression symptoms etc. If there is a site available where people , teen age kids can find their relevant books, read the reviews , see the ratings then it will be easier for them to buy the kinds of books they want. After reading books they can also share their opinions, reviews which will help and encourage others to read more books.

Solution

The Book Review Website is enabling readers to get to know appropriate books and wide ranges of relevant information about them. This offers users to choose books either according to category and deep dive on them. Or users can get recommendations from the system based on their previous history or relevancy.

How to Run

Website

The website is uploaded on AWS and can be accessed online anytime on the following link

<https://main.d32xukquy2hn8i.amplifyapp.com/>

As the UI is adaptable to screen sizes , the topbar may look different based on screen resolution. Please zoom in/ out browser accordingly to see and adjust to the best viewing experience).

Running Locally

The project is running on React Framework (More on it later). To run it locally the code can be downloaded from the github repository , create a react project with the name `book-review` and copy the github project content to the newly created project and run accordingly. Here are the steps-

- Create a new react app locally. Here are the instructions from the official website to do that using different npm. In the commands the app name is `my-app` , this should be replaced with the project name `book-review` .
- Download the sources from the Github public [repository](#) and paste them inside the newly created project.
- Here are the [scripts](#) to run the project locally from the React Developer website..

Project Architecture

FrontEnd

This project is built with [React](#) framework (A javascript library to build user interface). To make the website work faster the whole website is actually a [single-page application](#). This resulted in a faster transition and a smooth website (More on this later on [User interface section](#)). React is ideal for this single page design. The programming part is implemented with [JSXlanguage](#) and design implemented by HTML and CSS.

The website always had a navigation bar on the top to navigate to any relevant page at any point of time. And for the body the code calculates current state and logic based on database items and user clicks/behavior to decide what component should be rendered at that point of time. So even though the website has multiple pages for showing book lists, existing reviews of a book , adding new reviews etc. - all these are rendered on a single page.

Here's code for the navigation bar component. This is mostly straightforward logic. Most of the buttons are static and just one-two options based on usertype. For other com

main → book-review / src / Navbar.js / <> Jump to ▾

Go to file ⋮

 **saida123-csueb** Suggestion implementation

Latest commit ab425b4 2 days ago ⏪ History

1 contributor

67 lines (57 sloc) | 2.89 KB

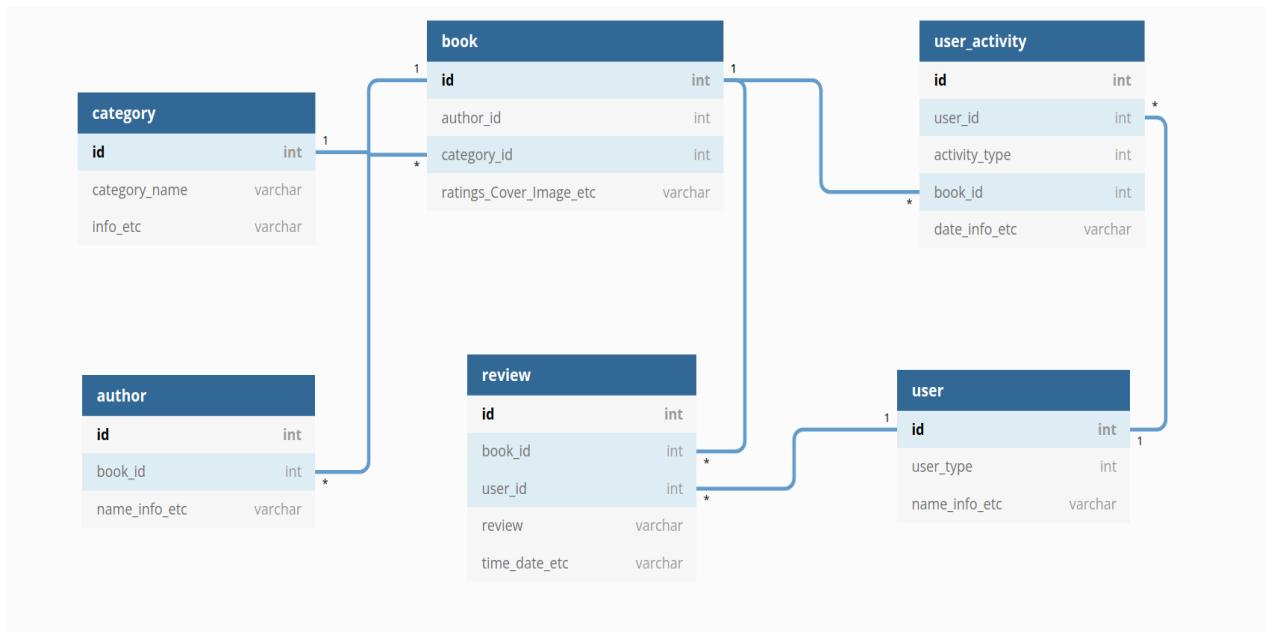
Raw Blame ⌂ ⌍ ⌒

```
1 import React from "react";
2
3 class Navbar extends React.Component {
4     constructor(props) {
5         super(props);
6         this.state = {searchBox: ''};
7         this.handleChange = this.handleChange.bind(this);
8         this.handleSubmit = this.handleSubmit.bind(this);
9     }
10    handleChange(event) {
11        this.setState({searchBox: event.target.value});
12    }
13
14    handleSubmit(event) {
15        if (!this.state.searchBox || /^\\s*$/.test(this.state.searchBox)) return;
16        event.preventDefault();
17        this.setState({searchBox: ''});
18        this.props.pageHandler('showbooks', this.state.searchBox);
19    }
20
21
22    showFavorite(){
23        if (this.props.userInfo && this.props.userInfo.type && (this.props.userInfo.type == 'user' || this.props.userInfo.type == 'admin')) {
24            return( <li onClick={() => this.props.pageHandler('showbooks', 'userSugg')}><a href="#">{this.props.userInfo.name}'s Suggestion</a></li> );
25        }
26    }
27
28    showLogin() {
29
30        if (this.props.userInfo && this.props.userInfo.type && (this.props.userInfo.type == 'user' || this.props.userInfo.type == 'admin') ) {
31            return(<li onClick={() => this.props.pageHandler('logout', '')}><a href="#">Log Out</a></li>);
32        } else {
33            return(<li onClick={() => this.props.pageHandler('login', '')}><a href="#">Log In</a></li>);
34        }
35
36    }
37
38    render() {
39        return(
40            <body class="navbody">
41                <nav>
42                    <div class = "icon" onClick={() => this.props.pageHandler('start', '')}> Book <b style={{color: 'red'}}>Review</b></div>
43                    <div class= "search_box">
44                        <form onSubmit={this.handleSubmit}>
45                            <input type = "search" placeholder="Search Here.." value={this.state.searchBox} onChange={this.handleChange}/>
46                            <button class= "fa fa-search" ></button>
47                        </form>
48                    </div>
49                    <ol>
50                        { this.showFavorite()}
51                        <li onClick={() => this.props.pageHandler('showbooks', 'Trending')}><a href="#">Trending</a></li>
52                        <li onClick={() => this.props.pageHandler('showbooks', 'Sci-fi')}><a href="#">Sci-Fi</a></li>
53                        <li onClick={() => this.props.pageHandler('showbooks', 'Thriller')}><a href="#">Thriller</a></li>
54                        <li onClick={() => this.props.pageHandler('showbooks', 'Drama')}><a href="#">Drama</a></li>
55                        <li onClick={() => this.props.pageHandler('showbooks', 'Non Fiction')}><a href="#">Non Fiction</a></li>
56                        <li onClick={() => this.props.pageHandler('showbooks', 'Children')}><a href="#">Children</a></li>
57                        <li onClick={() => this.props.pageHandler('showbooks', 'Other')}><a href="#">Other</a></li>
58                        {this.showLogin()}
59                    </ol>
60                </nav>
61            </body>
62        );
63    }
64
65 }
66
67 export default Navbar;
```

Database

Design

The database of this project is PostgreSQL. It's a free and open-source relational database management system emphasizing extensibility and SQL compliance. As it's a relational database all data are organized in different tables and they are connected by reference (or foreign key in some notation). Here is the basic design structure



How this design structure is used to implement the search and recommendation will be discussed in a later chapter.

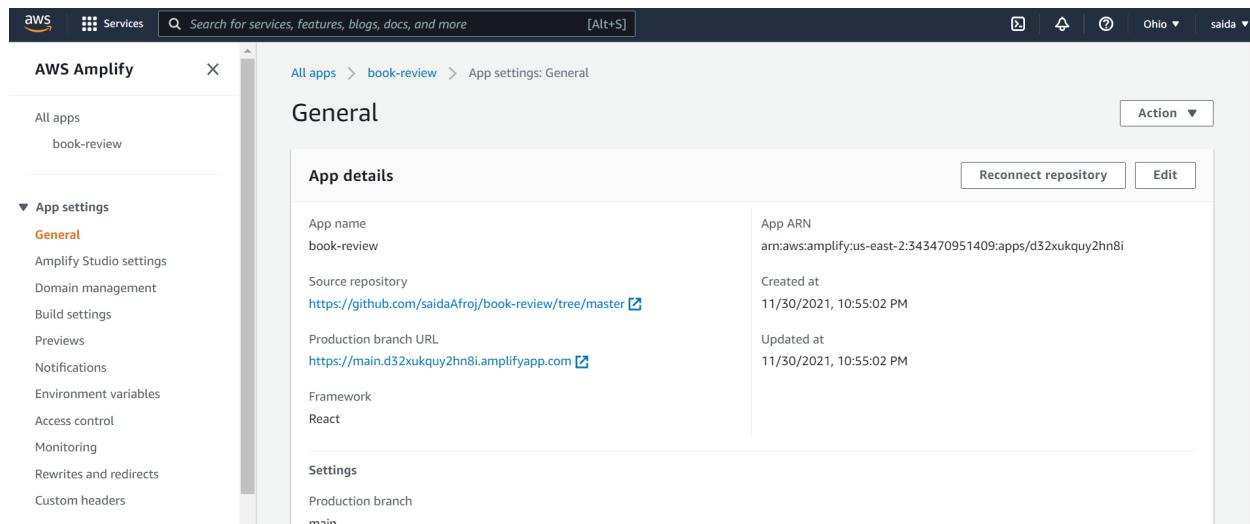
Why Relational database instead of NoSQL

Generally for thread/comment/discussion systems NoSQL databases are faster and more ideal. But with this review website I used PostgreSQL which is a modern SQL database. In this design Book table , review tables are placed separately. This is fine for this use case. Because for the initial search, recommendation and exploration stages only relevant books are shown to users based on rating , user etc. Reviews are not fetched then. Whenever a user selects a book then only their reviews are displayed. So at that point only reviews of those books are fetched from the database. This is a simple and fast Query. Complex queries where tables are joined to get data are mostly not needed. So the regular SQL database is more appropriate for this use case and design and implementation is much simpler.

Cloud Configuration

The whole website is uploaded on AWS (Amazon Web Services). The service that has been used here is [AWS Amplify](#). AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve.

For React projects AWS amplify directly fetches underlying code from any popular code repository (in this case Github) and creates a new version automatically after some time. So it reduces the time to launch code significantly. Once AWS Amplify is set up and given access to a github repository , then most of the work happens behind the eyes. Below is the general setting on the site's service. It was created at 11/30 and last updated on 11/30. But changes that are pushed onto GitHub are also present in the service.



The screenshot shows the AWS Amplify console interface. On the left, there's a sidebar with 'All apps' and 'book-review'. Under 'App settings', 'General' is selected. The main area shows the 'General' tab for the 'book-review' app. The 'App details' section contains the following information:

Setting	Value
App name	book-review
Source repository	https://github.com/saidaAfroj/book-review/tree/master
Production branch URL	https://main.d32xukquy2hn8i.amplifyapp.com
Framework	React
Settings	Production branch main

At the top right, there are 'Reconnect repository' and 'Edit' buttons. The top navigation bar includes 'Services', a search bar, and account information for Ohio.

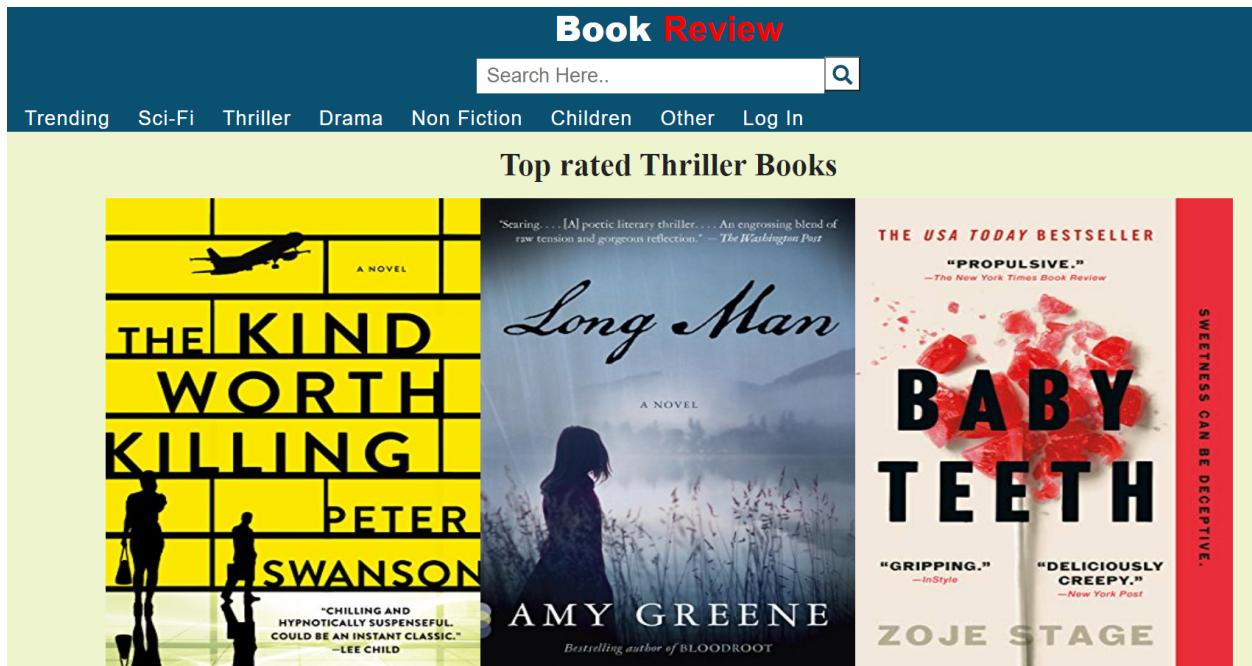
Implementation & Features

User Interface and User experience

One of the primary goals of this website is to give readers a fast and snappy experience with minimal effort. The single page react based website already gives a smooth experience. Some notable features of the website from UX point of view:

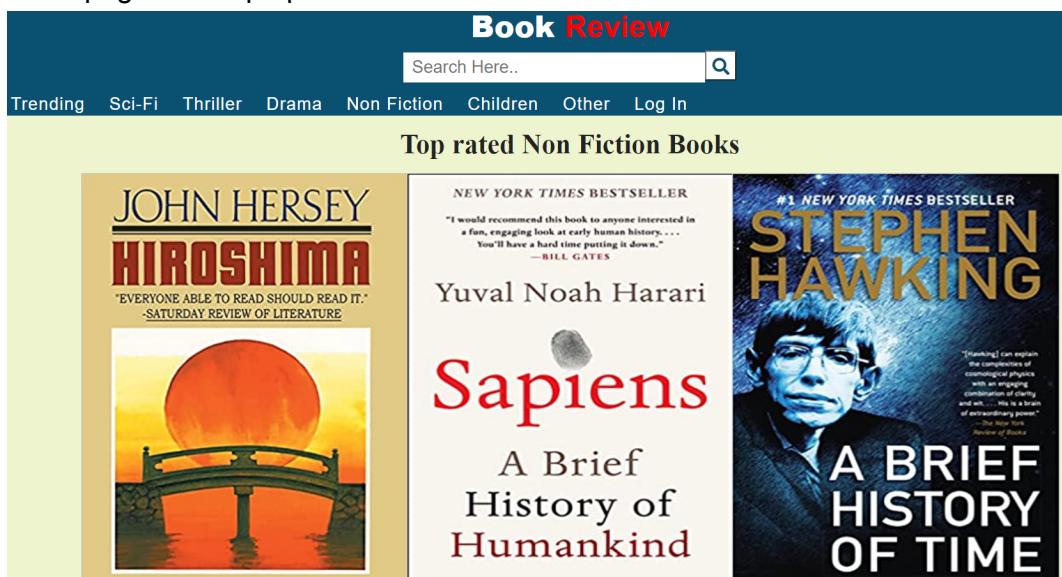
Organized by categories

Books organized by major categories so the reader can deep dive on some categories. Also an always present navigation bar easily allows the user to move between different categories. Here is a screenshot from the site

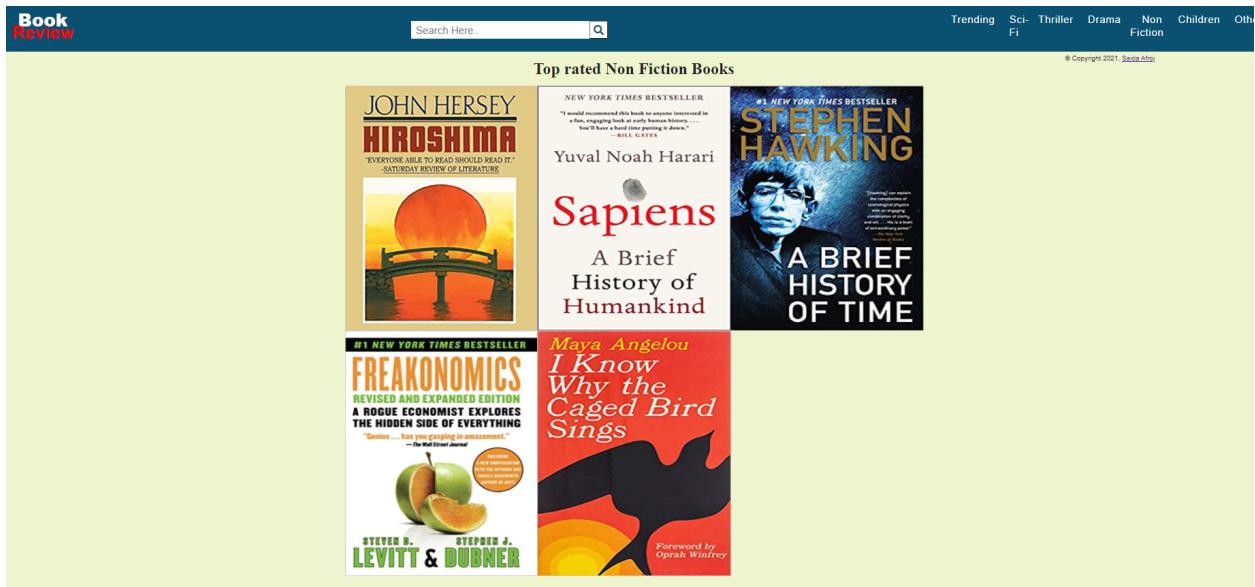


Adjustable UI for screen sizes

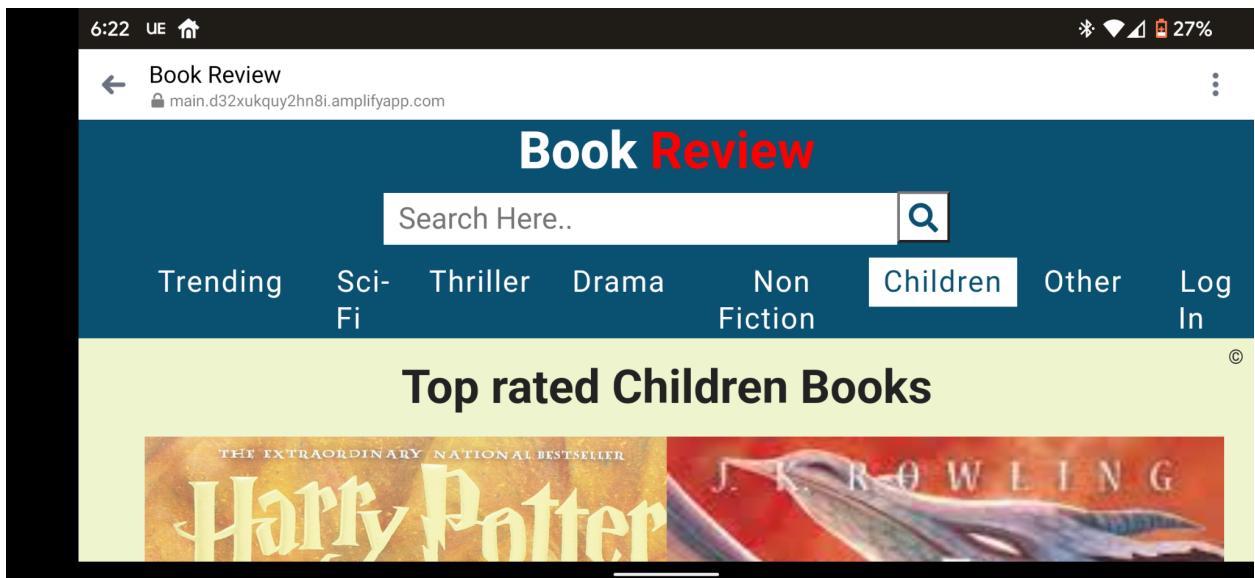
The interface in the website adjusts according to the screen sizes. Here's the regular view of the home page on a laptop screen:

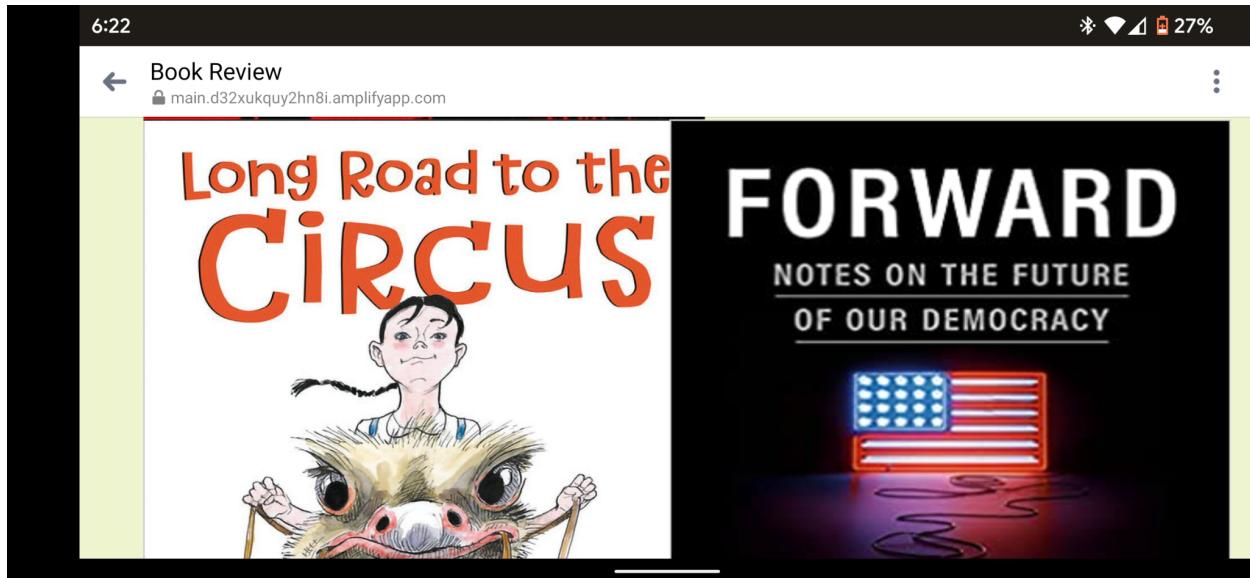


Now for high resolution big screens the navbar adjusts for a different view. (The same experience can be also be observed by adjusting zoom percentages in a browser)



Also this website is currently created only for the web. But even then it's still usable from a mobile display browser. Here's two screenshots from a 5 inch display mobile browser





Responsive Book Display Grid

In the navigation pages the book cover is shown in a big noticeable and attractive manner. Especially for young users , the big cover may attract them. Relevant results are shown on a grid. These grids are responsive and on hover they provide more info for the user. Here in this screenshot the cursor is hovered on top of the “Hiroshima” book, and it displayed related info

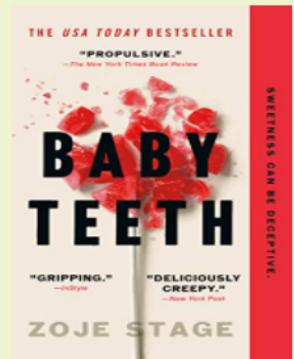
Advanced CSS

The animation of the website is implemented by CSS. This makes it very fast and compatible with all modern browsers and all clients.

Smooth Review Page

Review page is also very simple and inline with the theme of the rest of the website.

Baby Teeth



Author: Zoje Stage
Category: Thriller
Rating: 4.1

Summary: This 2018 novel follows a dysfunctional family with a terror of a daughter (you might call her a bad seed). Hanna is seven years old and has never spoken, but that's the least of her mother's worries; you'd probably feel the same if your little girl were trying to kill you.

 [Boris Salinas](#) -Posted 6 months ago

This novel is a magical story that reveals the essence of life, love and loneliness. Due to the simplicity of the prose style, breezy and funny illustrations, vividness of imagination this simple yet insightful story is considered to be one of the greatest literary works of all times.

 [Aqib Norton](#) -Posted 5 months ago

One of the most striking features of the book is the uncompromising approach of its author. She constantly rejects all halftones and any attempts to evade from responsibility. Everything becomes very clear and straightforward: there is a way to creativity and if someone refuses to choose it, he betrays creativity and himself. This position requires taking a very big responsibility and only few people can bear it.

 [Alishba Blake](#) -Posted 4 weeks ago

...

truth from fiction is eroding. We can no longer assume that fundamental functions of American democracy, like the smooth counting of votes on Election Day or the ability of Congress to pass laws, will occur. Some of us have stopped believing in science, while others have simply come to doubt the possibility that brighter days lie ahead. The unprecedented disruption of the COVID-19 pandemic laid our anxieties bare. Unity and consensus seem like fading dreams.



[Matteo Donovan](#) -Posted 2 weeks ago

The suspense keeps the pages flying, but what sets this one apart is the palpable sense of onrushing doom.



[Elizabeth Gilbert](#) -Posted a week ago

WRITING STYLE: 3.5/5 , SUBJECT: 4/5, CANDIDNESS: 4.5/5, RELEVANCE: 3.5/5, ENTERTAINMENT QUOTIENT: 3.5/5



[Kaiya Peck](#) -Posted 4 days ago

While I definitely recommend this book to other readers, I would recommend it to older teenagers, mainly because it will resonate better with them. The writing is tame enough that younger teens could also read it, but most of the characters are adults or on the verge of adulthood. Older readers would take the most from it since they can not only relate, but they may also better pick up on and appreciate Cassandra's sometimes subtle humor.

Write a review

[ADD REVIEW](#)

© Copyright 2021, [Saida Afroj](#)

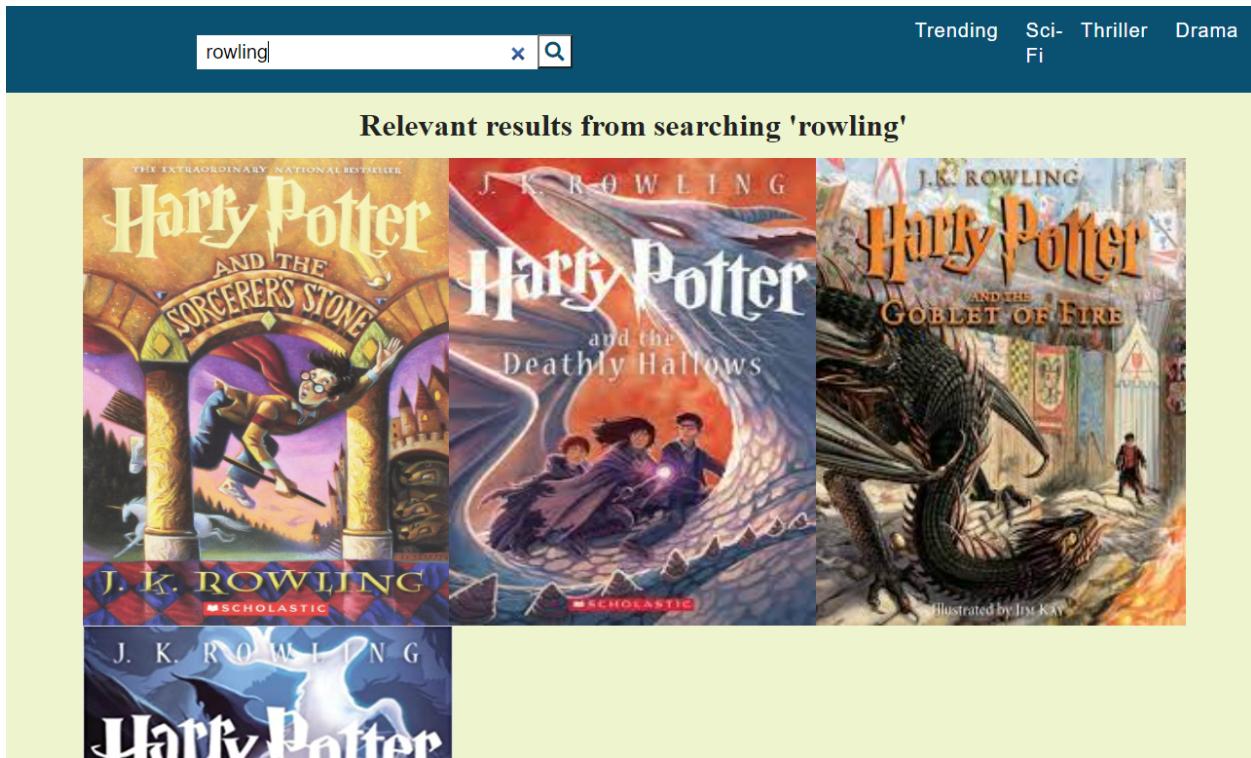
Search & Recommendation

Main purpose of the website is to suggest to users to find books which are likely to be interesting to them. So intelligent searches and recommendations play a vital role in this website. This site has many advanced features to help users find what they are looking for.

Advanced Search

Users can explore any category of book by selecting through the navigation bar. But searching from the search bar and finding books at the front gives the user a magical experience. This advanced search also enabled me to create a simpler website. Such as - separate pages for author based books suggestion are not needed as a single search with author's name can fulfill

that need. Here a sample search with author's name (Rowling returned all the harry potter books from the database) -



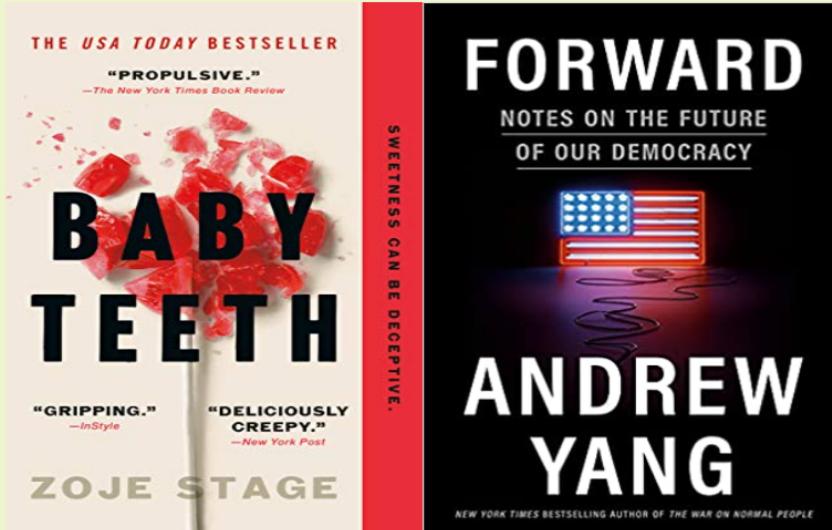
Search in the search bar not only looks at the title/authors name , it also digs deep into books with more detailed info and reviews to find proper suggestions. Suppose here searching for 'bestseller' shows two books that are actually bestsellers.

bestseller

x 

Trending Sci-Fi

Relevant results from searching 'bestseller'



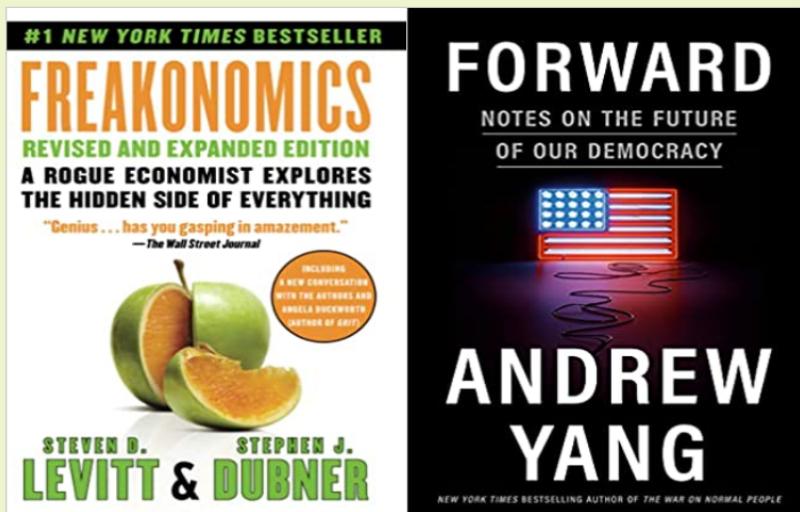
Here searching economics also resulted in books which are related books.

economid

x 

Trending Sci-Fi Thriller

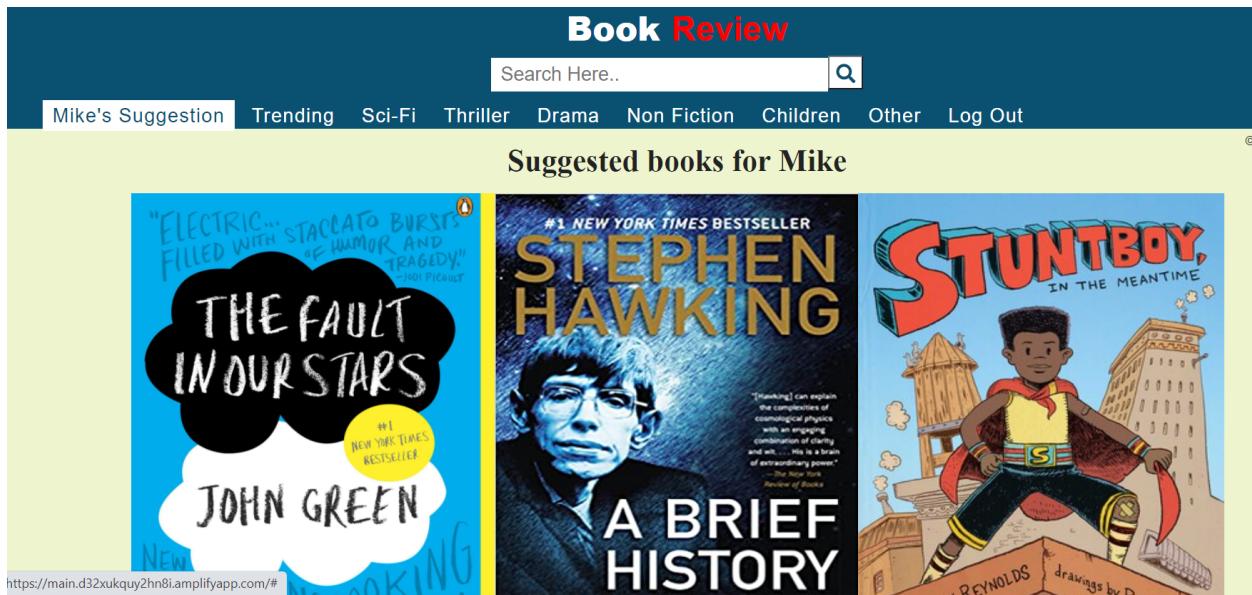
Relevant results from searching 'economic'



More can be done to improve search results when there is enough data to train a model. More on this in the later [chapter](#).

User Based Recommendation

After a user logs in, another page shows up like this which is that user's suggestion. Here is a screenshot suggested page when user is Mike -

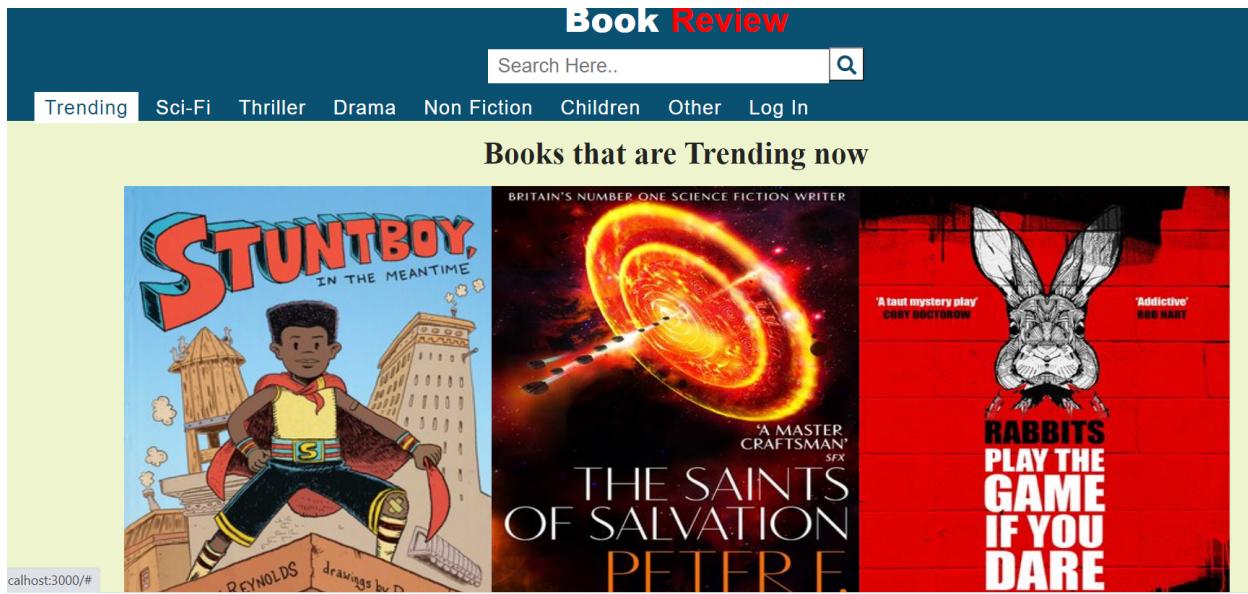


This page shows recommendations based on user history. In the database there's have a user table. This table collects which books the user visited and which books they send review or done any other activity. Thus it suggests appropriate books for the user.

Currently this is a heuristic based suggestion. In the future there's a plan to add friends list with their activity history. When there will be enough data to train a machine learning model it can also suggest magical suggestions to users. More on it later.

Trends

And also based on visit count from all users , there is a trending page which suggests users for the most trending books at that time.



Current Cloud Version Limitation

Because of time and resource constraints, the website uploaded at AWS has a limited version of the database. Some dummy data has been pushed for the user history and activity log. That's why the cloud version is missing some features.

And also in the cloud one admin doesn't have the capability to add new books yet. This is currently turned off for the cloud one as book cover image file uploading is not implemented yet here.

Future Improvements

There are some improvements which will increase the websites quality and relevancy by a big margin. Each of these works require significant time and effort which was not possible to give before the current due date.

Friend List with Activity Logs

For each user the database is keeping track of their own activity log. Enabling sharing and connectivity among multiple users will enable many features for the future. A timeline or activity feed of friends will help users to get to know what their friends/peers are interested about. It will also improve suggestions & recommendations.

Machine Learning Models

After gathering a significant amount of data related to user activity history, it will be possible to turn them into a feature for machine learning models. These models will also be able to train and verify itself with more data in the future and provide a more magical experience for the user.

Automated Data Collection

Currently Reviews for books are only reviews which are given in the website. For popular and new books , it'll be possible to fetch their review from publicly available websites. Amazon Books, Google Books and ISBN provide public API and they can be used for this purpose.

Important Links

- [Project on cloud Link](#)
- [GitHub code repository link](#)
- [Presentation Video](#)