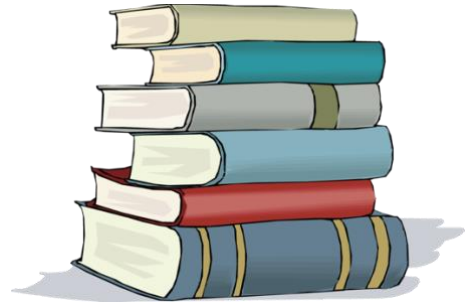


# *Web & Mobile Development Programming Project: (CS 5590)*

## *Library Management System*



**Team Name:** Team 5

**Team Leader:** Avinash Ganguri (Class ID: 11)

**Team Members:** Bindu Gadiparthi (Class ID: 10)

Jagadheesh Maraju (Class ID: 19)

Mehmet Acikgoz (Class ID: 1)

## Story of the project: (Problem Statement)

People going to an actual library for just the sake of checking a book availability or renewing a book is a process that should be minimized. During Covid-19, maintaining social distance and accessing such features through online is the safest way.

Moreover, library management is a troublesome process in maintaining all the records manually with the quantity list and their issue, renewal dates. Through an online application, it makes safer and easier way for all the Students and Staff in accessing and maintaining book services.



Either a bibliophile or a bookworm, anyone who wants to read a book can use this application for ease, instead of going to an actual library for certain features.

### Who

**Lucy** is a busy 23-year old graduate studying in University of Missouri at Kansas City who lives in Overland Park, KS. He works part time as a student lab assistant and manages his time between assignments, exams and work. He lives alone and has many other duties to fill in a household. In his free time, he spends time with friends, reading books and watching movies.

### What

Lucy has a difficulty in managing time between all his activities and he's focusing on to improve his productivity. He's trying to simplify his lifestyle and also being safe by not traveling for simple things. Being a bibliophile, he reads a lot of books and doesn't want to take extra time to go to an actual library for checking book availability or renewing a book.

### When

After Covid-19, people are trying to be safe by staying in home and avoiding traveling for simple tasks. Lucy one among them also focusing on improving his productivity

where such services being online can be a lot helpful and time saving process in the modern era for everyone.

## **Where**

Community of Overland Park, Kansas. Within the city limits there are only a few libraries and many out of them are closed due to Covid-19 restrictions. And Lucy staying far from his university and it is quite difficult for him to travel all along the way from Overland Park to Kansas City for just renewing a book. And also, for the staff who works in library, managing all the books and their date of issues and renewals in manual record format is quite difficult.

However, if Lucy and the staff at UMKC face this problem, we could wonder how many people or schools face such difficulty as this problem.

## **Why**

Managing books and their renewals is a tiresome process and there might be a chance of errors. Through an online application portal, it is easy for library management for both the Students and Staff with proper and accurate details to access certain features. Moreover, a portal where users can search books with different categories and genres, find other user reviews on the particular book or comment their own review for it, changing renewal status and many such features can be done at ease with an application rather than a manual library system.

There are people like Lucy who wants to maximize their productivity with ease and safe environment.



## **Data that helps to develop the project:**

### **Who**

The dataset we're using is a manually created dataset that includes information about all of the books present in the UMKC library, and it's aimed at undergraduate and graduate students from various backgrounds. We use Google Books API for certain features such as book image or book description as well. The dataset's fields reflect the story's key characteristics.



## What

Our dataset includes many records with an estimated of 8 columns, two of which will be changed during the runtime. The names of the columns, as well as their descriptions, are mentioned below.

**Title:** The title or name given to the book.

**Author:** The name of the author.

**Description:** About the book in a detailed way.

**Genre:** The category of the book.

**Location:** The location of the book will be updated in this column.

**Rating:** The review of the book can be viewed in this column.

**Quantity:** The number of books taken will be updated in this column.

**Issue date:** On the day the book is published, this column is updated with the current date.

## When

During Spring 2021 term, the data will be obtained from the UMKC library website. The information gathered is cross-sectional. The information reflects a series of books released between 2009 and 2021 in the UMKC library.

**Where**

There are no geographical details in this dataset. The information in the dataset pertains to the library's selection of books for undergraduate and graduate students at UMKC.

**Why**

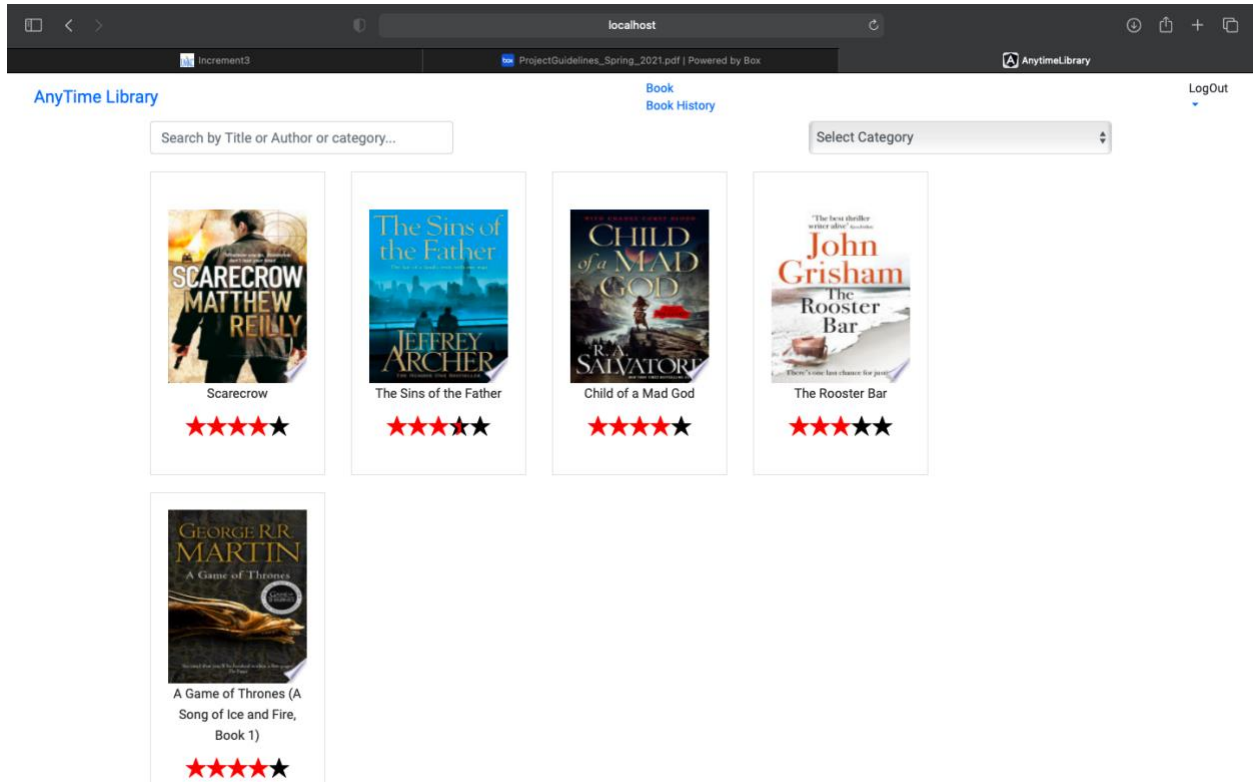
The data will be collected for a better understanding of the book availability that are present in the university and also helps students to get a glance of their availability and ease of renewals. Also, helps the staff to manage the book inventory without any hassle.

**How**

We've added two new columns to the table: Quantity and Issue date, which are changed when a book is published and returned. The changes done will be updated automatically.

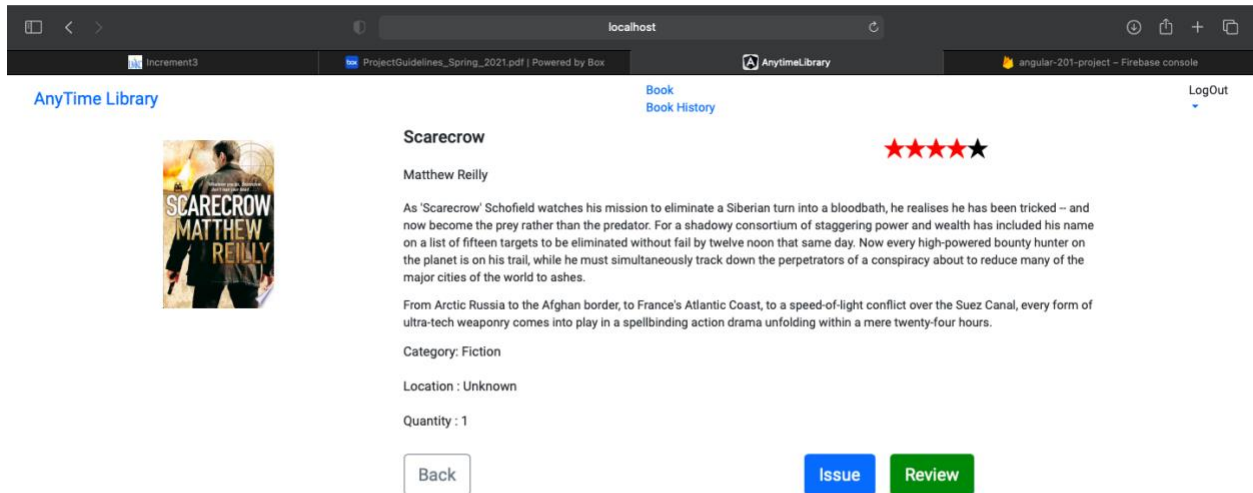
## Working Screens from the project:

This is the Home page of our Application – **AnyTime Library** (*Library Management System*)



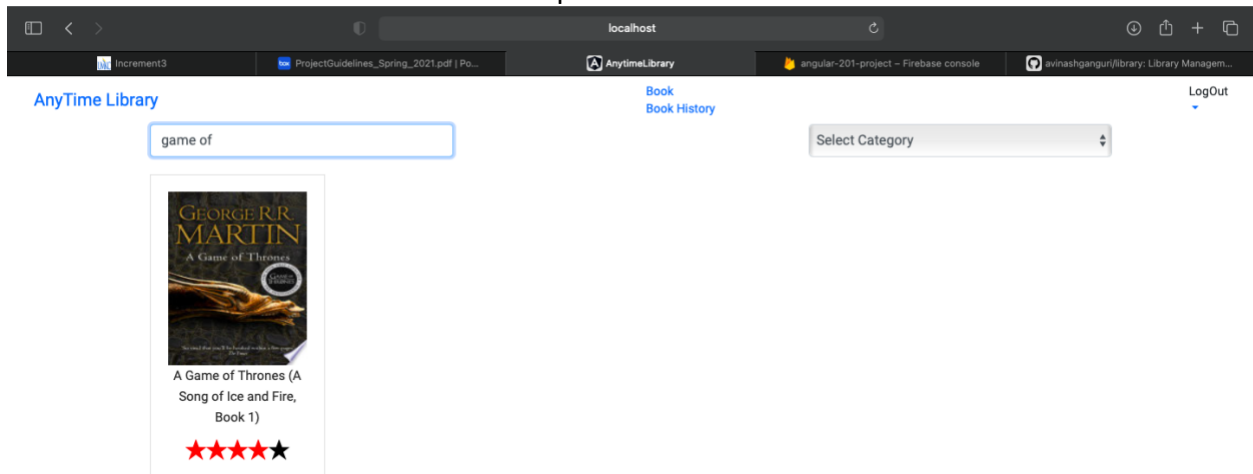
Where all the books present in the library are displayed here with the user ratings.

When the user selects a book, he can see the details of it including Category, Location and the book quantity.



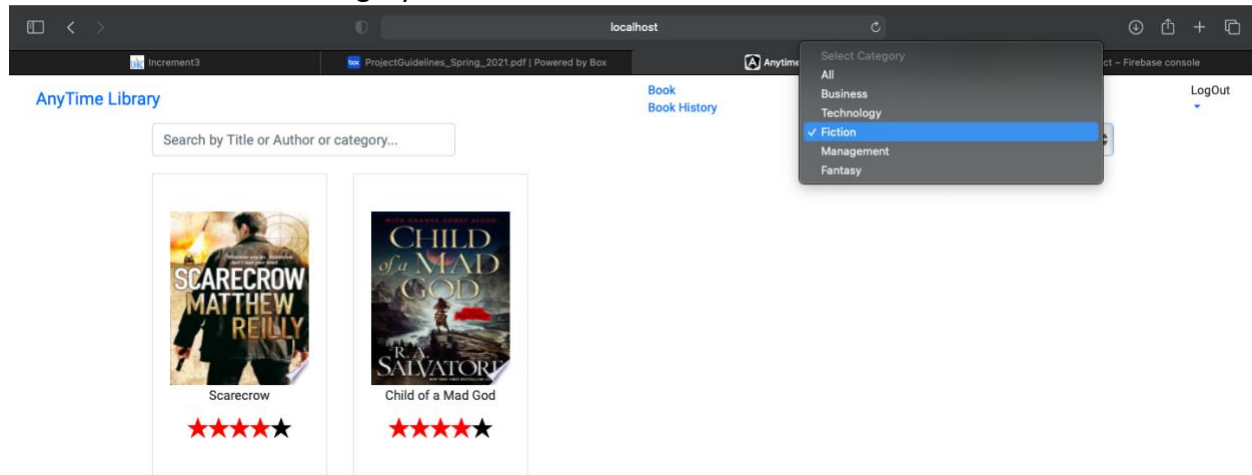
And here you can have the option of Issue and Review the book. The Review is still under development and currently users can issue the book in the increment 3.

User can search book from the search bar provided.

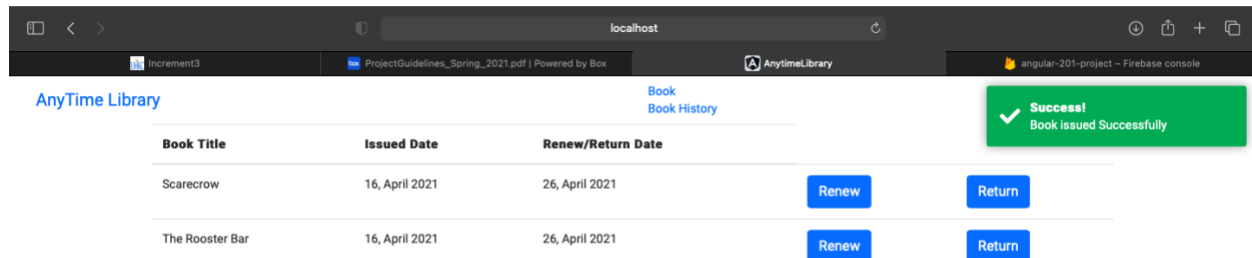




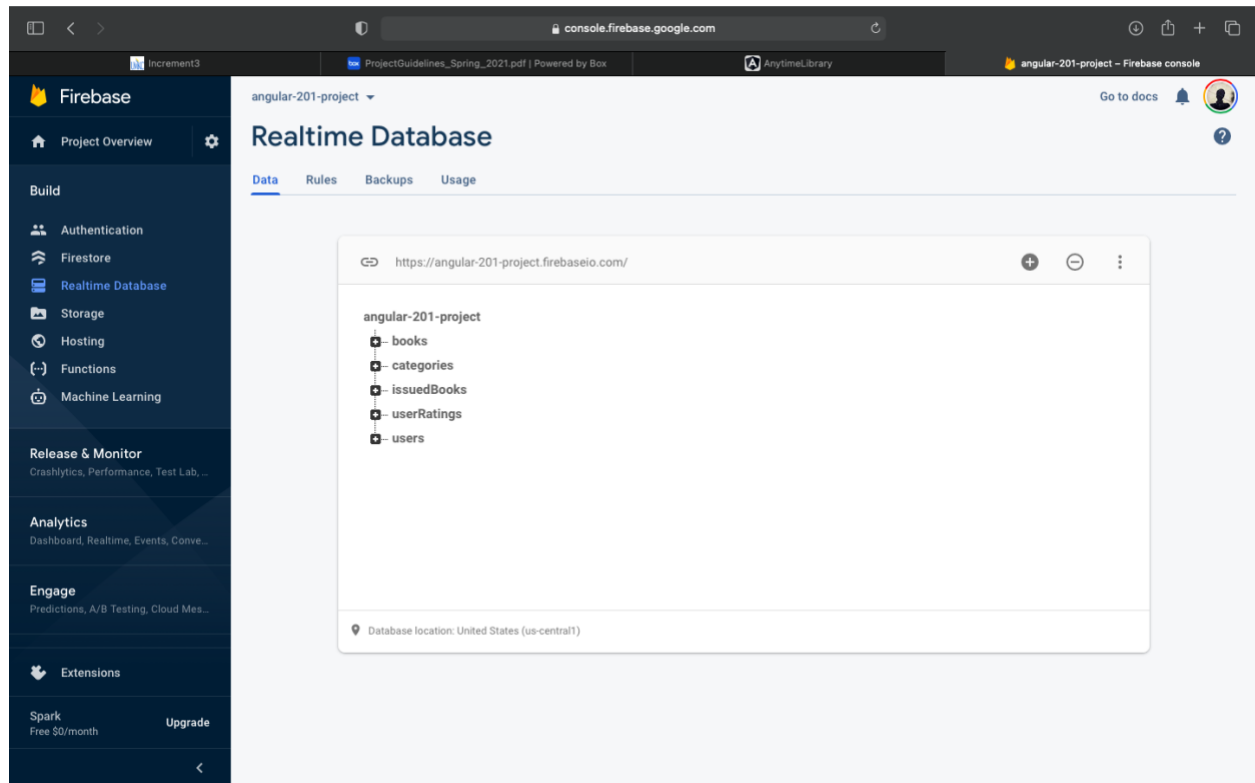
Also, user can select books based on different categories provided. And books are filtered based on the selected category.



After issuing the books, they are added to the book history queue page where a user can renew or return the book.



We have used Firebase Realtime Database for our project based on its different functionalities and analytic features.



## For Increment-4 (Updated)



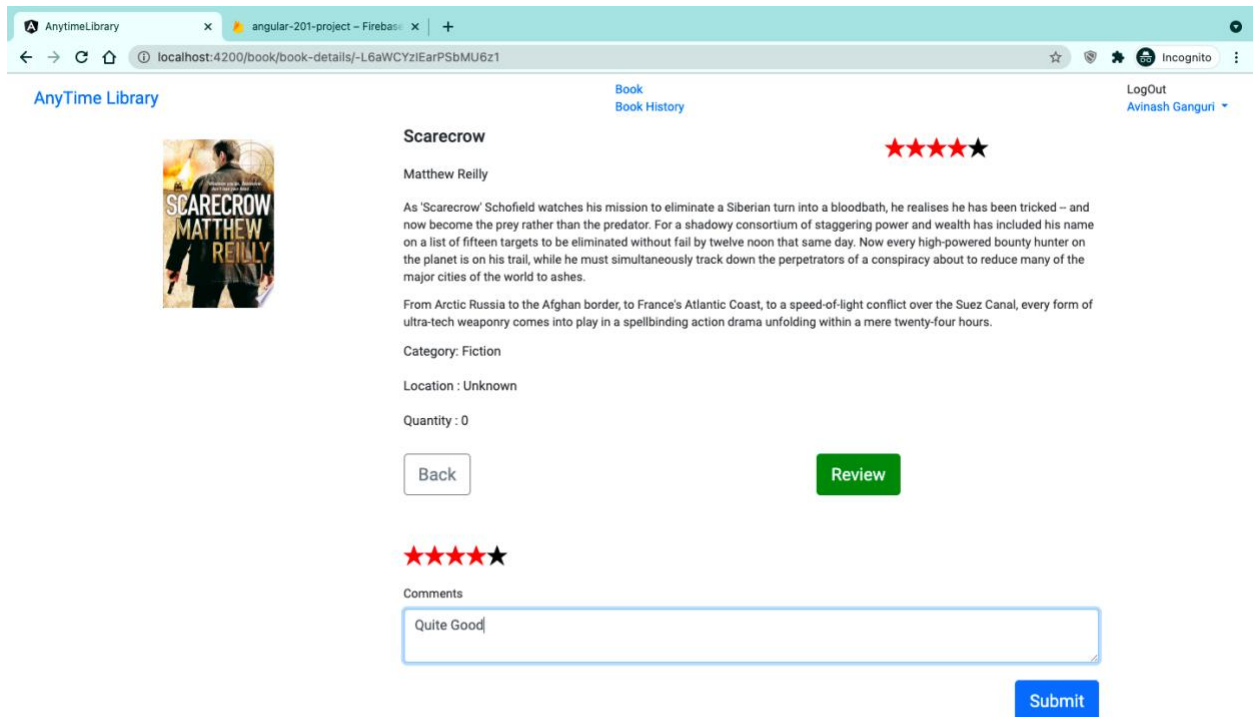
AnyTime Library [SignUp/Login](#)

E-Mail


Password

[Login](#) | [Switch To Sign Up](#)

Now users can give a review and comment as well for the books.



AnyTime Library [Book](#) [Book History](#) [LogOut Avinash Ganguri](#)



**Scarecrow**

Matthew Reilly

★★★★★

As 'Scarecrow' Schofield watches his mission to eliminate a Siberian turn into a bloodbath, he realises he has been tricked – and now become the prey rather than the predator. For a shadowy consortium of staggering power and wealth has included his name on a list of fifteen targets to be eliminated without fail by twelve noon that same day. Now every high-powered bounty hunter on the planet is on his trail, while he must simultaneously track down the perpetrators of a conspiracy about to reduce many of the major cities of the world to ashes.

From Arctic Russia to the Afghan border, to France's Atlantic Coast, to a speed-of-light conflict over the Suez Canal, every form of ultra-tech weaponry comes into play in a spellbinding action drama unfolding within a mere twenty-four hours.

Category: Fiction

Location : Unknown

Quantity : 0

[Back](#) [Review](#)

★★★★★

Comments

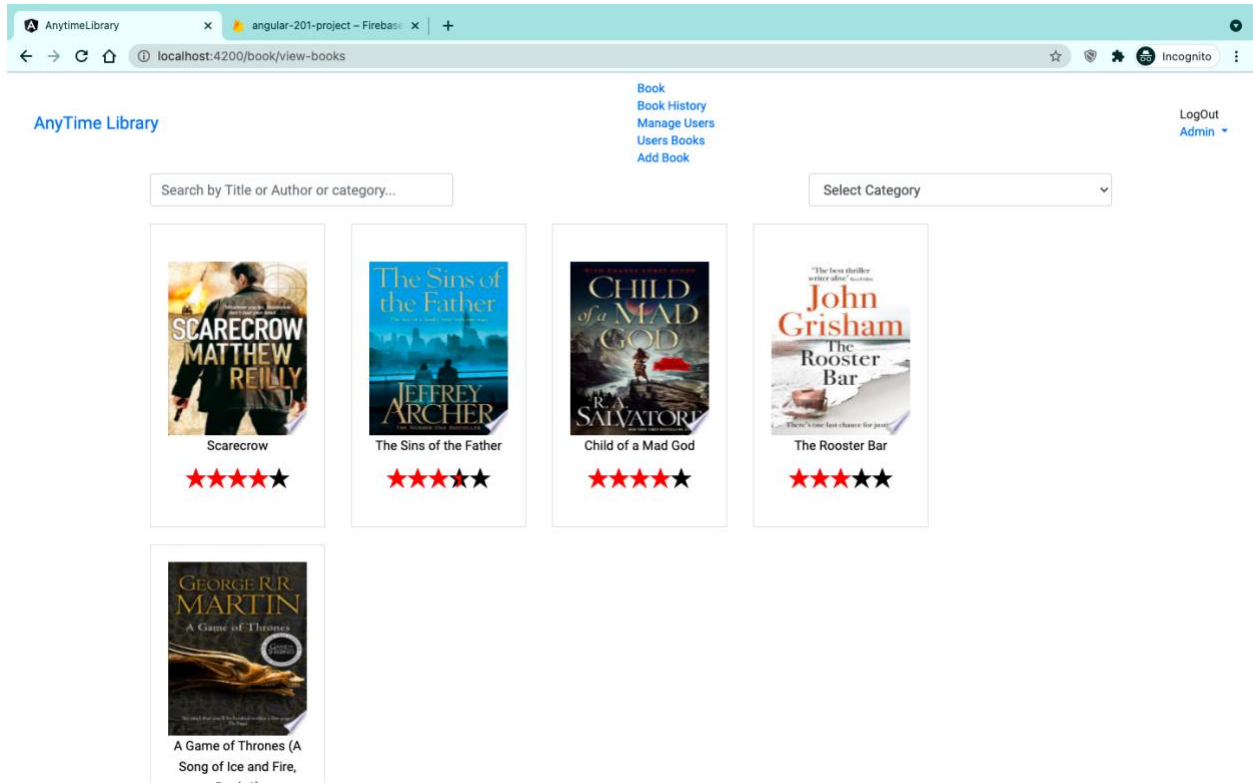
Quite Good

[Submit](#)

Admin has a lot of different functionalities and can be seen here,

He can

- Add or update Users
- Check on Users Books
- Add or update a New Book



Here you can observe that he can add a new user and also can edit current users.

AnyTime Library

Book  
Book History  
Manage Users  
Users Books  
Add Book

LogOut  
Admin

Add User

Name	Role	
Avinash Ganguri	User	Edit
Admin	Admin	Edit
Jagadeesh Maraju	User	Edit

Adding a new user,

AnyTime Library

Book  
Book History  
Manage Users  
Users Books  
Add Book

LogOut  
Admin

Name

EmailId

Role

Favourite Genres

Submit

Updating a current user,

AnytimeLibrary x angular-201-project - Firebase x +

localhost:4200/users/update-user/avinash@gmail.com

AnyTime Library

- Book
- Book History
- Manage Users
- Users Books
- Add Book

LogOut Admin

Name

Avinash Ganguri

EmailId

avinash@gmail.com

Role

User

Favourite Genres

Fiction

Submit

He can also view the books that are issued to different users,

AnytimeLibrary x angular-201-project - Firebase x +

localhost:4200/users/view-user-books

AnyTime Library

- Book
- Book History
- Manage Users
- Users Books
- Add Book

LogOut Admin

Search by Title or Author or category...

Avinash Ganguri

Admin

Jagadeesh Maraju

CHILD of a MAD GOD R. A. SALVATORE

Child of a Mad God

Return By: 20, May 2021

SCARECROW MATTHEW REILLY

Scarecrow

Return By: 20, May 2021

Adding a new book through ISBN, here the book details can be retrived by using Google Books API. And also if a particular book is not present or can't be added through using ISBN, then he can add the book details manully as well.

AnytimeLibrary

angular-201-project - Firebas

+

localhost:4200/book/add-book

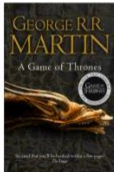
☆

Incognito

+

Add Book

Search



Title

Author(s)

Description

<p>HBO's hit series A GAME OF THRONES is based on George R. R. Martin's internationally bestselling series A SONG OF ICE AND FIRE, the greatest fantasy epic of the modern age. A GAME OF THRONES is the first volume in the series.</p>  
<p>Summers span decades. Winter can last a lifetime. And the struggle for the Iron Throne has begun.</p>  
<p>As Warden of the north, Lord Eddard Stark counts it a curse when King Robert bestows on him the office of the Hand. His honour weighs him down at court where a true man does what he will, not what he must ... and a dead enemy is a thing of beauty.

ImageUrl

Location

Category

Quantity

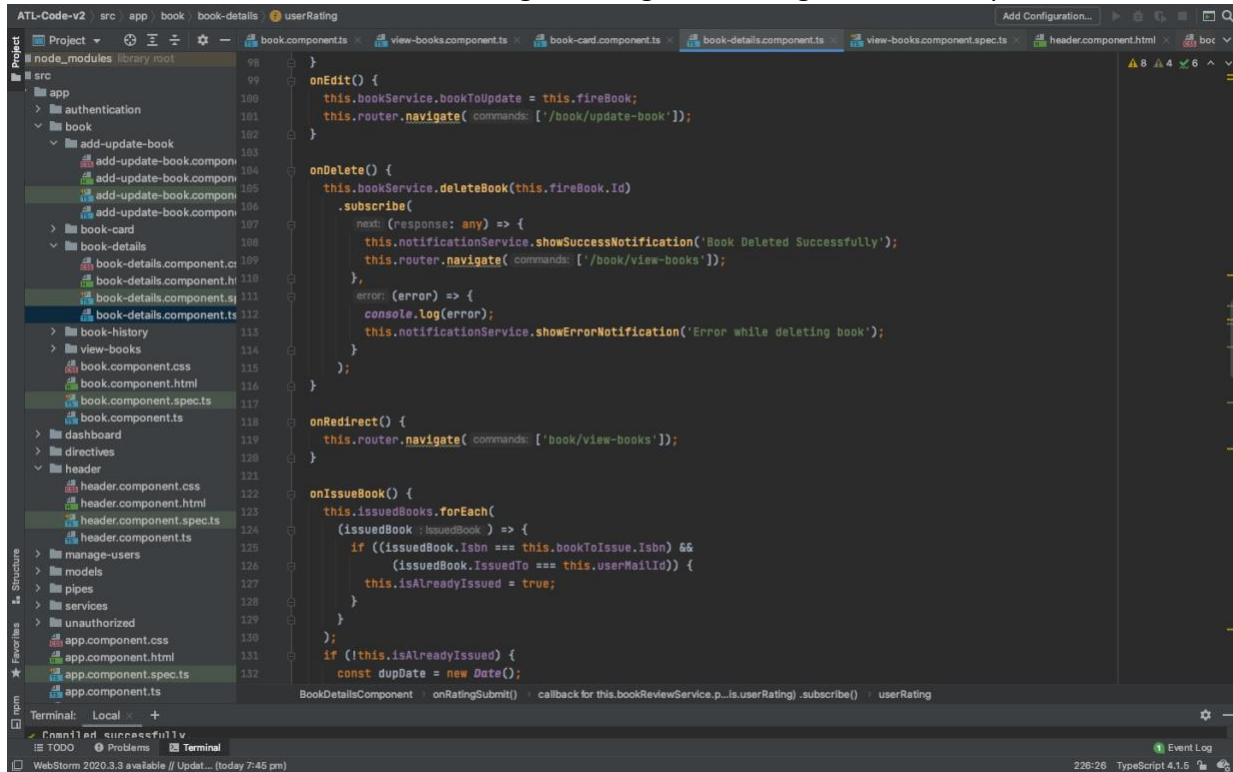
Average Rating

Back

Submit

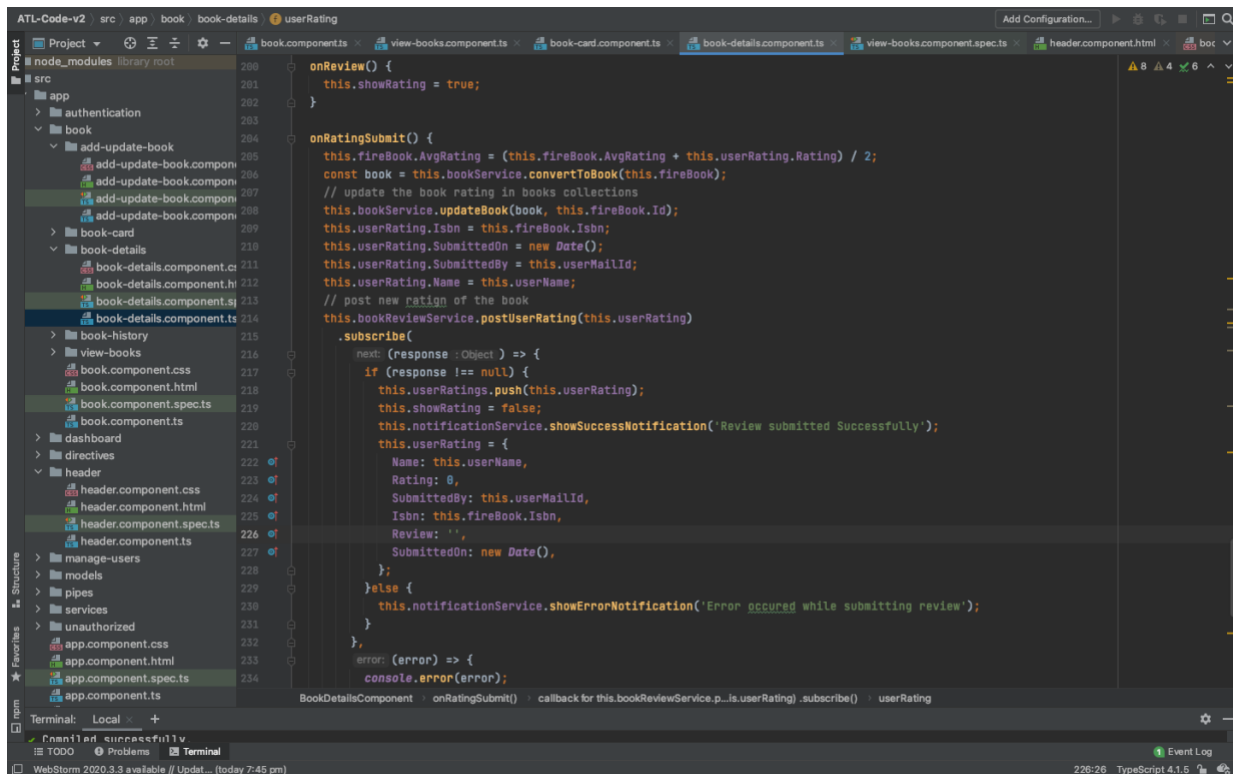
## Important code snippets

This is the code where it used for editing, deleting, redirecting a book takes place.



```
ATL-Code-v2 | src | app | book | book-details | userRating
Project | node_modules | library | root
src
  app
    authentication
    book
      add-update-book
        add-update-book.component.css
        add-update-book.component.html
        add-update-book.component.spec.ts
        add-update-book.component.ts
      book-card
      book-details
        book-details.component.css
        book-details.component.html
        book-details.component.spec.ts
        book-details.component.ts
      book-history
      view-books
        book.component.css
        book.component.html
        book.component.spec.ts
        book.component.ts
      dashboard
      directives
      header
        header.component.css
        header.component.html
        header.component.spec.ts
        header.component.ts
      manage-users
      models
      pipes
      services
      unauthorized
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
  Structure
  Favorites
  Terminal: Local
  Compiled successfully
  TODO
  Problems
  Terminal
  WebStorm 2020.3.3 available // Update... (today 7:45 pm)
  226:26 | TypeScript 4.1.5 | Event Log
```

```
98 }
99
100 onEdit() {
101   this.bookService.bookToUpdate = this.fireBook;
102   this.router.navigate( commands: ['/book/update-book'] );
103 }
104
105 onDelete() {
106   this.bookService.deleteBook(this.fireBook.Id)
107   .subscribe(
108     next: (response: any) => {
109       this.notificationService.showSuccessNotification('Book Deleted Successfully');
110       this.router.navigate( commands: ['/book/view-books'] );
111     },
112     error: (error) => {
113       console.log(error);
114       this.notificationService.showErrorNotification('Error while deleting book');
115     }
116   );
117 }
118
119 onRedirect() {
120   this.router.navigate( commands: ['book/view-books'] );
121 }
122
123 onIssueBook() {
124   this.issuedBooks.forEach(
125     (issuedBook: IssuedBook) => {
126       if ((issuedBook.Isnb === this.bookToIssue.Isnb) &&
127         (issuedBook.IssuedTo === this.userMailId)) {
128         this.isAlreadyIssued = true;
129       }
130     }
131   );
132   if (!this.isAlreadyIssued) {
133     const dupDate = new Date();
134     // ... (rest of the code) ...
135   }
136 }
```

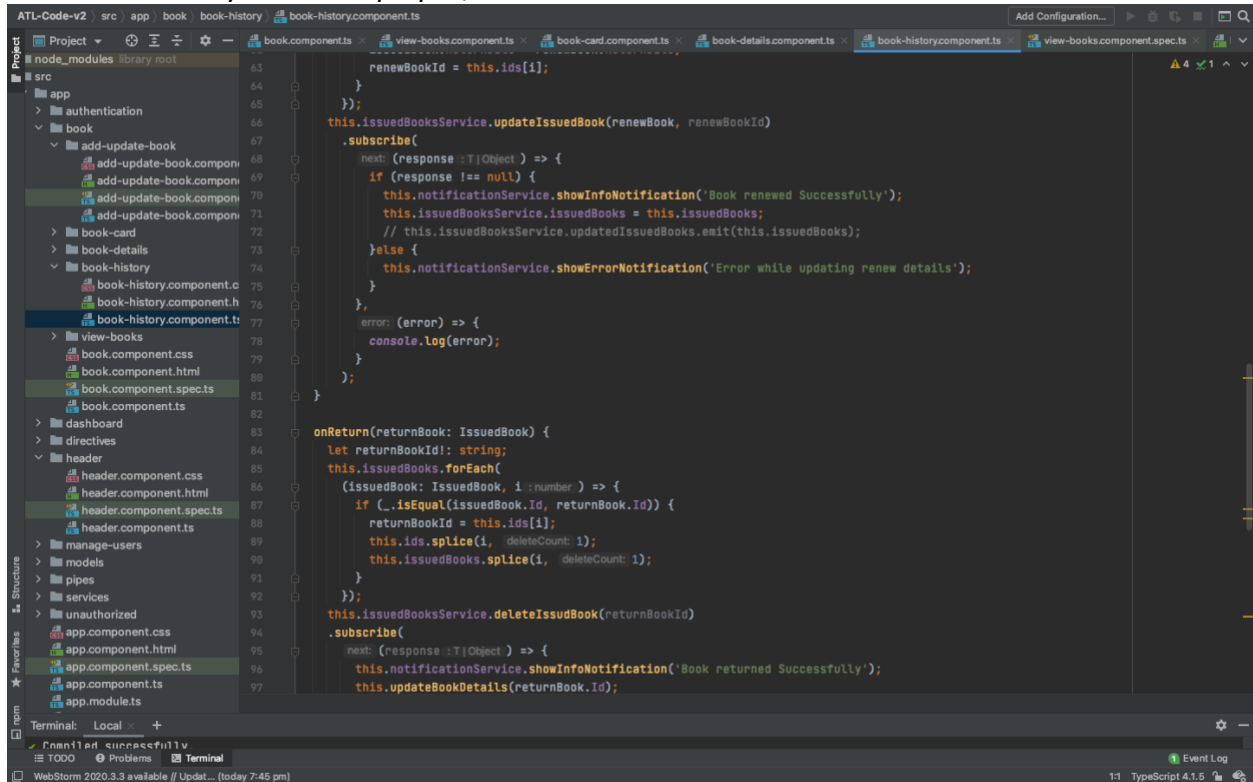


```
ATL-Code-v2 | src | app | book | book-details | userRating
Project | node_modules | library | root
src
  app
    authentication
    book
      add-update-book
        add-update-book.component.css
        add-update-book.component.html
        add-update-book.component.spec.ts
        add-update-book.component.ts
      book-card
      book-details
        book-details.component.css
        book-details.component.html
        book-details.component.spec.ts
        book-details.component.ts
      book-history
      view-books
        book.component.css
        book.component.html
        book.component.spec.ts
        book.component.ts
      dashboard
      directives
      header
        header.component.css
        header.component.html
        header.component.spec.ts
        header.component.ts
      manage-users
      models
      pipes
      services
      unauthorized
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
  Structure
  Favorites
  Terminal: Local
  Compiled successfully
  TODO
  Problems
  Terminal
  WebStorm 2020.3.3 available // Update... (today 7:45 pm)
  226:26 | TypeScript 4.1.5 | Event Log
```

```
200 onReview() {
201   this.showRating = true;
202 }
203
204 onRatingSubmit() {
205   this.fireBook.AvgRating = (this.fireBook.AvgRating + this.userRating.Rating) / 2;
206   const book = this.bookService.convertToBook(this.fireBook);
207   // update the book rating in books collections
208   this.bookService.updateBook(book, this.fireBook.Id);
209   this.userRating.Isbn = this.fireBook.Isbn;
210   this.userRating.SubmittedOn = new Date();
211   this.userRating.SubmittedBy = this.userMailId;
212   this.userRating.Name = this.userName;
213   // post new rating of the book
214   this.bookReviewService.postUserRating(this.userRating)
215   .subscribe(
216     next: (response: Object) => {
217       if (response != null) {
218         this.userRatings.push(this.userRating);
219         this.showRating = false;
220         this.notificationService.showSuccessNotification('Review submitted Successfully');
221         this.userRating = {
222           Name: this.userName,
223           Rating: 0,
224           SubmittedBy: this.userMailId,
225           Isbn: this.fireBook.Isbn,
226           Review: '',
227           SubmittedOn: new Date(),
228         };
229       } else {
230         this.notificationService.showErrorNotification('Error occured while submitting review');
231       }
232     },
233     error: (error) => {
234       console.error(error);
235     }
236   );
237 }
```

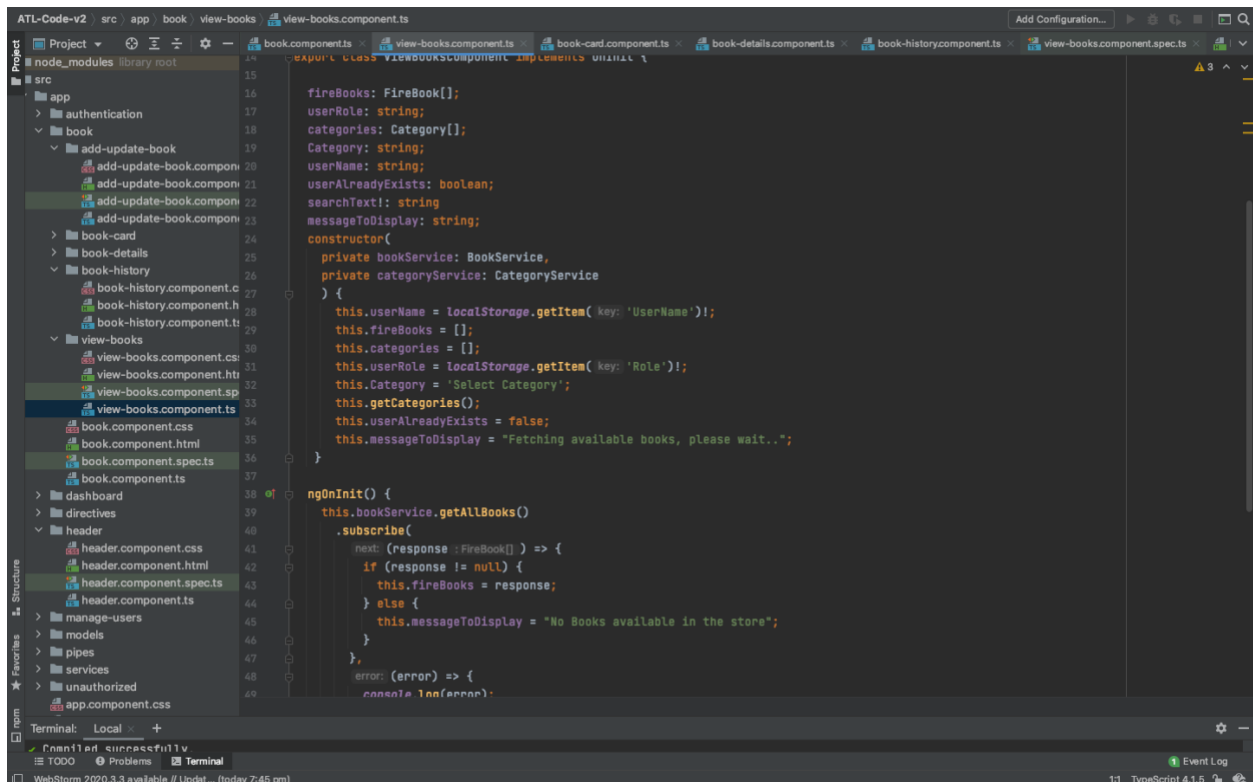


Here book history can be displayed,



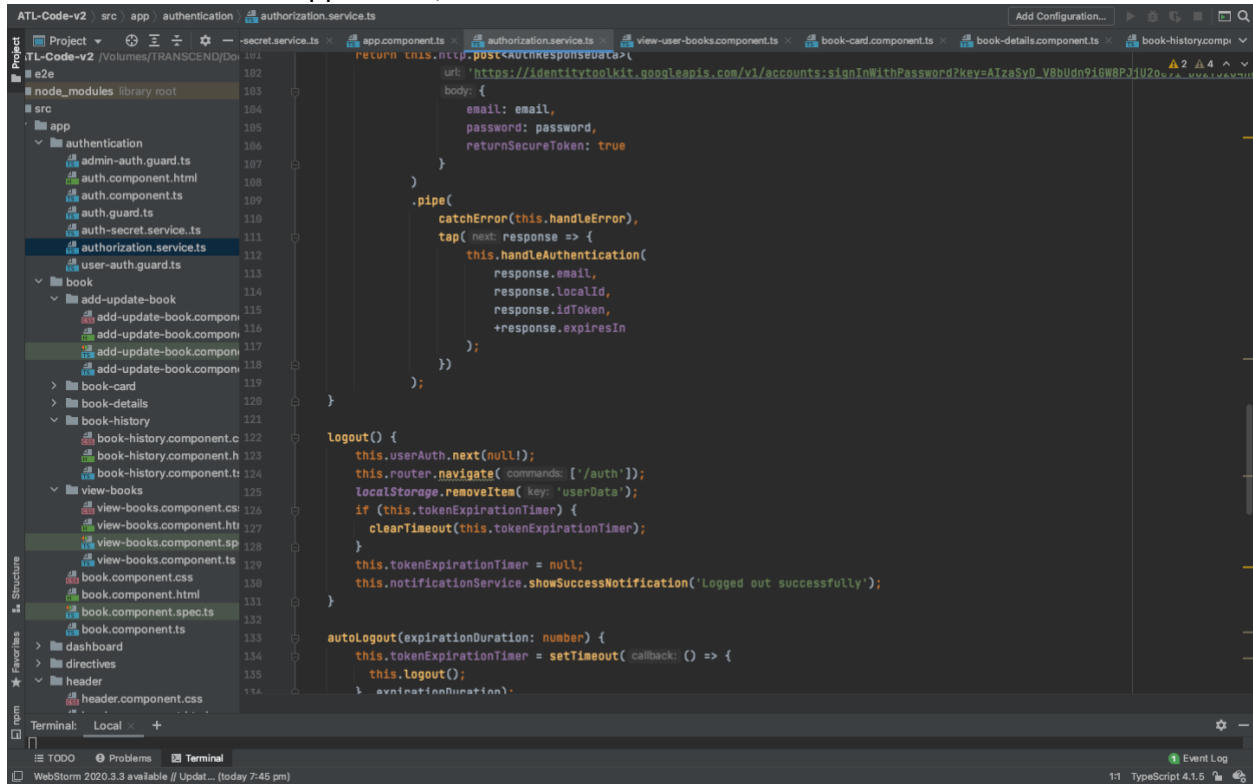
```
63     renewBookId = this.ids[i];
64   }
65   });
66   this.issuedBooksService.updateIssuedBook(renewBook, renewBookId)
67   .subscribe(
68     next: (response :T|Object) => {
69       if (response !== null) {
70         this.notificationService.showInfoNotification('Book renewed Successfully');
71         this.issuedBooksService.issuedBooks = this.issuedBooks;
72         // this.issuedBooksService.updatedIssuedBooks.emit(this.issuedBooks);
73       } else {
74         this.notificationService.showErrorNotification('Error while updating renew details');
75       }
76     },
77     error: (error) => {
78       console.log(error);
79     }
80   );
81 }
82
83 onReturn(returnBook: IssuedBook) {
84   let returnBookId!: string;
85   this.issuedBooks.forEach(
86     (issuedBook: IssuedBook, i :number) => {
87       if (!_.isEqual(issuedBook.Id, returnBook.Id)) {
88         returnBookId = this.ids[i];
89         this.ids.splice(i, deleteCount: 1);
90         this.issuedBooks.splice(i, deleteCount: 1);
91       }
92     });
93   this.issuedBooksService.deleteIssuedBook(returnBookId)
94   .subscribe(
95     next: (response :T|Object) => {
96       this.notificationService.showInfoNotification('Book returned Successfully');
97       this.updateBookDetails(returnBook.Id);
98     }
99   );
100 }
```

Book availability can be seen here,

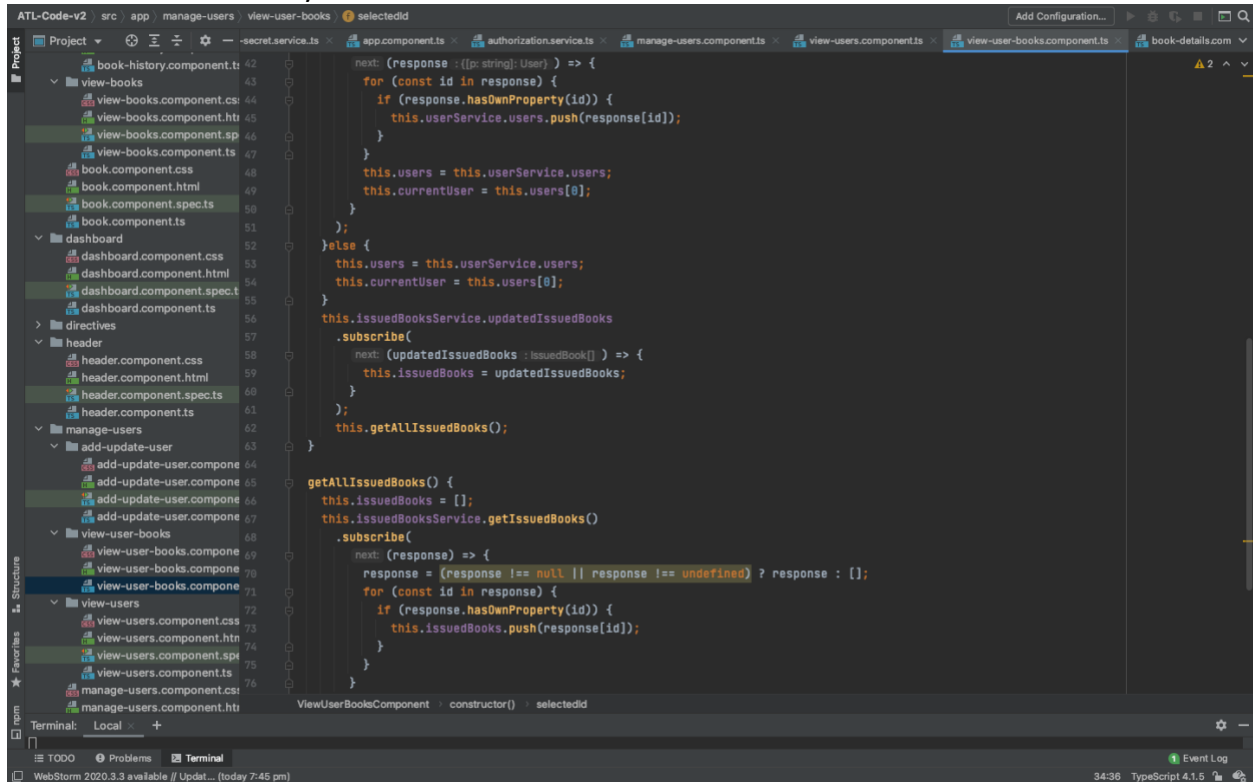


```
14 @export class ViewBooksComponent implements OnInit {
15
16   fireBooks: FireBook[];
17   userRole: string;
18   categories: Category[];
19   Category: string;
20   userName: string;
21   userAlreadyExists: boolean;
22   searchText!: string;
23   messageToDisplay: string;
24   constructor(
25     private bookService: BookService,
26     private categoryService: CategoryService
27   ) {
28     this.userName = localStorage.getItem( key: 'UserName')!;
29     this.fireBooks = [];
30     this.categories = [];
31     this.userRole = localStorage.getItem( key: 'Role')!;
32     this.Category = 'Select Category';
33     this.getCategories();
34     this.userAlreadyExists = false;
35     this.messageToDisplay = "Fetching available books, please wait..";
36   }
37
38   ngOnInit() {
39     this.bookService.getAllBooks()
40     .subscribe(
41       next: (response : FireBook[] ) => {
42         if (response !== null) {
43           this.fireBooks = response;
44         } else {
45           this.messageToDisplay = "No Books available in the store";
46         }
47       },
48       error: (error) => {
49         console.log(error);
50       }
51     );
52   }
53 }
```

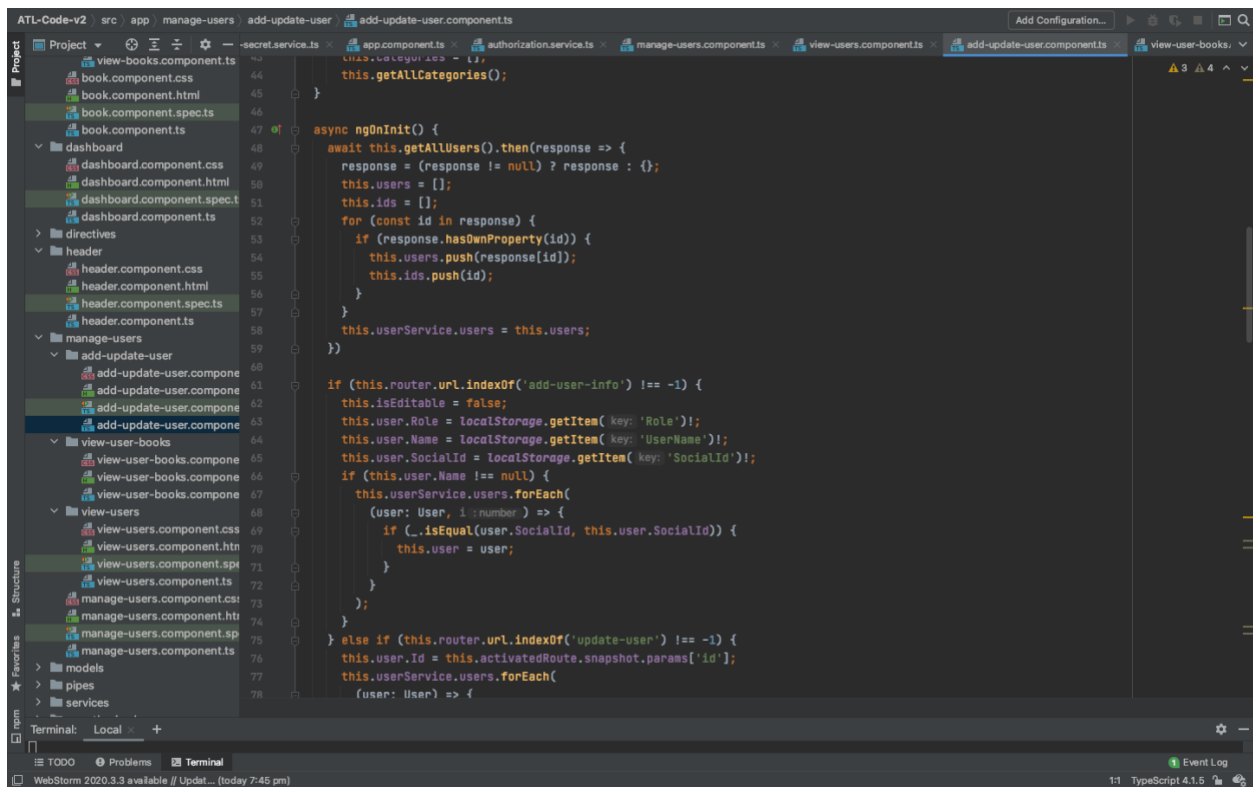
## Authentication of the application,



## View users book history - Admin



Adding or updating users -Admin,



```
44      this.getAllCategories();
45    }
46  }
47  of
48  async ngOnInit() {
49    await this.getAllUsers().then(response => {
50      response = (response != null) ? response : {};
51      this.users = [];
52      this.ids = [];
53      for (const id in response) {
54        if (response.hasOwnProperty(id)) {
55          this.users.push(response[id]);
56          this.ids.push(id);
57        }
58      }
59      this.userService.users = this.users;
60    });
61  }
62  if (this.router.url.indexOf('add-user-info') !== -1) {
63    this.isEditable = false;
64    this.user.Role = localStorage.getItem('Role')!;
65    this.user.Name = localStorage.getItem('UserName')!;
66    this.user.SocialId = localStorage.getItem('SocialId')!;
67    if (this.user.Name !== null) {
68      this.userService.users.forEach(
69        (user: User, i: number) => {
70          if (!_.isEqual(user.SocialId, this.user.SocialId)) {
71            this.user = user;
72          }
73        }
74      );
75    }
76  } else if (this.router.url.indexOf('update-user') !== -1) {
77    this.user.Id = this.activatedRoute.snapshot.params['id'];
78    this.userService.users.forEach(
79      (user: User) => {
```

Terminal: Local x +

WebStorm 2020.3.3 available for Update (today 7:45 pm)

1:1 TypeScript 4.1.5

## **Work Sharing between Teammates:**

Avinash Ganguri – 25%  
Jagadeesh Maraju – 25%  
Bindu Gadiparthi – 25%  
Mehmet Acikgoz – 25%

Bindu and Mehmet worked together in building the User Interface for the project and worked on adding the book issue queues and added authentication features to the application.

Avinash and Jagadeesh worked together in adding backend functionalities to the project by managing the firebase database and worked on admin functionalities for the project.

## **Issues with the project:**

Storing and Retrieving data with angular from firebase is a new concept which we felt quite difficult at the starting and picked upon later.

Updating the book quantity after a user issues a book is a difficult feature we faced to add. Got references from internet and solved it.

User Reviews are a quite important feature in our library management system, and it is not yet functional saving for each user. It is the issue we solved in the final increment.

**GitHub link for the project:** <https://github.com/avinashganguri/library>

**Video link for the project:**

<https://umkc.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3a05c1ec-4d3e-4b30-918f-ad25004bec8c>

**Presentation link for the project:** [https://drive.google.com/file/d/1w\\_u\\_46Fq-Bu637bWGlaKNwjPrAZgqNwB/view?usp=sharing](https://drive.google.com/file/d/1w_u_46Fq-Bu637bWGlaKNwjPrAZgqNwB/view?usp=sharing)

## **References:**

<https://umkc.app.box.com/s/vj9vi3k0nzv1tgvubrp807wq8k1jxxzw>

<https://umkc.app.box.com/s/by9k956lz1fh67u4y4b18ndtqbexgbkw>

<https://csestudyzone.blogspot.com/2019/04/uml-case-study-library-management-system.html>

[https://www.sheffield.ac.uk/polopoly\\_fs/1.440722!/file/HowtoWriteaProblemStatement.pdf](https://www.sheffield.ac.uk/polopoly_fs/1.440722!/file/HowtoWriteaProblemStatement.pdf)

<https://www.youtube.com/watch?v=TucRRB57fi8>