BOOKSEEKER

by Thenmozhi S

Submission date: 13-May-2021 08:55AM (UTC+0530)
Submission ID: 1584880971
File name: Bookseeker_report.docx (7.43M)
Word count: 4005
Character count: 22778

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

6th Semester Project Report on

BOOKSEEKER

Submitted by

ANURAG ROY
PES1201802356

Jan – May, 2021

under the guidance of

Internal Guide
Dr. S Thenmozhi
Associate Professor
Department of Computer Applications,
PESU, Bengaluru – 560085

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER APPLICATIONS
PROGRAM – MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled

BOOKSEEKER

is a bonafide work carried out by

ANURAG ROY-PES1201802356

in partial fulfillment for the completion of 6th semester project work in the Program of Study MCA with specialization in Web Technology under rules and regulations of PES University, Bengaluru during the period Jan. 2021 – May 2021. The project report has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Internal Guide
Dr. S Thenmozhi
Associate Professor
Department of Computer Applications,
PES University, Bengaluru - 560085

Chairperson
Dr. Veena S

Dean-Faculty of Engineering& Technology
Dr. B K Keshavan

Name and Signature of Examiners:

Examiner 1: *Examiner 2:* *Examiner 3:*

DECLARATION

I, **Anurag Roy,** hereby declare that the project entitled, ***Bookseeker,*** is an original work done by us under the guidance of **Dr. S Thenmozhi** Designation, Affiliation, and is being submitted in partial fulfillment of the requirements for completion of 6th Semester course work in the Program of Study **MCA**. All corrections/suggestions indicated for internal assessment have been incorporated in the report. The plagiarism check has been done for the report and is below the given threshold.

PLACE: Bangalore
DATE:

ANURAG ROY
PES1201802356

ACKNOWLEDGEMENT

This project would not have been successful without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I express my deep sense of gratitude to the Vice-Chancellor, PES University, Dr. J Suryaprasad and Chairperson, Department of Computer Applications, Dr. Veena S, for providing the platform and opportunity to work on the project.

I am highly indebted to Dr. S Thenmozhi, Associate Professor, PES University for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project.

I would like to express my gratitude towards my parents for their kind cooperation and encouragement which helped me in the completion of this project.

My thanks and appreciation to my friends for the constant support and people who have willingly helped me out in this project.

ANURAG ROY
PES1201802356

ABSTRACT

Bookseeker is a web-based electronic commerce application that provides a unified platform for the sale of hard copy and softy copy books. The user is able to pick up books based on recommendations, views on particular books. Feedback can be provided to the user as per the user's express reading. The reader can enjoy the benefits of the subscription plan when purchasing or downloading books. The user can also reward the author in accordance with his book. In addition, readers are able to compare books to other readers. Hence, this application is a unique solution for book lovers.

CONTENTS

LIST OF FIGURE

LIST OF TABLES

INTRODUCTION

1.INTODUCTION

1.1 PROBLEM DESCRIPTION

Assume that if the user wants to buy a book first he will visit the shop or search the Internet and download it online. Suppose users will get both options of the book like hardcopy or softcopy in one platform, making life easy. Here is a bookseeker is a unique platform to buy books like a paper copy or electronic copy with the regional price. There are types of book enthusiasts who would love to read the book to gain knowledge or read the book to pass the time. This application has a variety of genres that help the reader search for books from the catalog by recommending books based on the reader search history. Some applications will not provide features like book recommendations, but in this application, the user can obtain recommended books and allow to provide a review for a specific book.

Users can purchase multiple books due to the features of the shopping cart in bookseeker. The reader can select a particular book with quantity and place the order. Since we have offered for registered users, the price will be less and good quality books.

Bookseeker is a unified web application which allow users to choose books based on user preference. Reader can pick book which reader like, buy books or download pdf file of book. Subscription plan will allow users to purchase and download books with discounts.

1.2 PROBLEM STATEMENT

Since there is no shared platform for book lovers where user can download and purchase books with less price and get books as per user choice. Book lovers who like to read books for knowledge and other book enthusiasts who like to read books to pass the time. So to overcome this problem user can use applications like BookSeeker.

1.3 PURPOSE

To help book lover to select book based on reader's choice. The reader can also provide reviews and rewards for authors and books written by them. The reader can refer books which the reader like to other readers.

1.4 SCOPE

This application allows users to download and purchase books as one platform. The reader can recommend books based on book title and provide reward for authors.

LITERATURE SURVEY

2. LITERATURE SURVEY

A literature survey provides us insights into a particular field by analyzing the problem. A literature survey includes existing solutions, methodologies, and historical contributions. It gives us a better understanding of the existing solutions as well as the idea of improvisations.

2.1 BACKROUND STUDY

Amazon

Amazon Books is a physical extension of Amazon.com. For twenty year's amazon is selling books online as hardcopy. This application delivers books in a different location at a fixed price. Discuss forum is also available as a service. Since Amazon has a variety of books users can search books based on the user's choice.

Kindle

Kindle is also a physical extension of Amazon.com. But kindle is only for selling hardcopy, softcopy of books. This application allows the user to purchase a softcopy of the book at a lesser price and can also read a sample of books for free later they buy if liked it. Users can buy books from a variety of genres by searching.

Librarything

Librarything is an application where users can purchase books as hardcopy. Users can provide a review of books based on the reader's experience and recommend books. Searching of books is possible based on user input. The only thing that is not possible in Librarything is the user cannot purchase a softcopy of books.

Shelfari

Shelfari is an application where the user interface is not that good or suitable for users. This application has only a hard copy of books and no option for trying a sample of books. Users cannot allow to gives a review for books. The catalog of books is not updated by the time.

Riffle

Riffle offers an easy and clean platform for users to store the book as library and user can search the book from the library as online. The reader may wish to refer to the book referred to by other readers.

Revish

Revish is a book-based social networking site where users can share their reviews of books after reading the books. The reader can only download soft copies of books from the website and searching for books is possible but the catalog is not vast so less variety of books is available.

2.2 PROPOSED SYSTEM

Bookseeker is a unified platform which allows book lovers to purchase and download of books based on user's preferences. Features of Bookseeker are – purchase of hardcopy and softcopy of books, searching catalog of the books, recommending based on the book title, review given by readers and subscription plan for readers or users, payment of books is also available.

2.1 TOOLS AND TECHNOLOGIES

Reactjs

This is an open-source JavaScript library used to build an application's user interface. The components made of using this library are reusable. Writing code with JSX is convenient. Fast rendering of components. It uses a JavaScript DOM object which is faster than a regular DOM. Easy to learn. The components provide better readability as well as easy to maintain the code. Since each component is independent of each other so that future development is also painless. React is a flexible, efficient, and declarative JavaScript library for building user interfaces.

Nodejs

Node.js is a cross-platform, open-source, JavaScript runtime environment that executes JavaScript code outside a web browser. Mostly used for server-side scripting. Using nodejs application fast, scalable network application. It is made on google V8 engine JavaScript. Using Nodejs application building fast, scalable network applications, as it's capable of handling a huge number of simultaneous connections with high throughput, which equates to high scalability. Gives the advantage of caching and results in highly scalable performances.

Momentjs

It is a free and open-source JavaScript library used for date and time. It is very easy to implement with the date object. It supports all type of time-zone around the world. It has multiple features, such as relative time, calendar time, durations, and multi-language support. It has an endless list of plugins that allow for additional features like time-zone support.

Mongoosejs

It is an object data modeling (ODM) library that provides a rigorous modeling environment for data, enforcing structure as needed while still maintaining the flexibility that makes MongoDB powerful. It provides us connectivity for Nodejs to MongoDB. It is an open-source library. It provides schema validation and apply relationship. Mongoose manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

Visual Studio Code

It is a free source-code editor made by Microsoft. It comprises multiple features that include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. VSCode is based on the Electron framework which is used to develop Node.js web apps that run on the Blink layout engine.

HARDWARE AND SOFTWARE REQUIREMENTS

3. HARDWARE AND SOFTWARE SPECIFICATION

3.1 HARDWARE REQUIREMENTS

Table 3.1 Hardware Requirements

HARDWARE	SPECIFICATIONS
PRIMARY MEMORY	6GB OR MORE
SECONDARY MEMORY	128GB OR MORE
PROCESSOR	Intel core i5

3.2 SOFTWARE REQUIREMENTS

Table 3.2 Software Requirements

PURPOSE	TOOLS AND TECHNOLOGIES
OPERATING SYSTEM	Windows 10
USER INTERFACE	React(17.02)
SERVER PROCESSING	Node.js(15.01)
DATABASE	MongoDB(4.4)
IDE	Visual Studio Code (1.54)

SOFTWARE REQUIREMENT SPECIFICATIONS

4. SOFTWARE REQUIREMENTS SPECIFICATIONS

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application.

4.1 USERS

ADMIN

Admin is a user which have access to approve the payment and subscription plan for every user.

AUTHOR

Author as a user have register in the website, then author have the access to add new books to the websites and display list of books.

USERS

Once the users log in the websites users can purchase hardcopy and softcopy of books. User can also search books based on book name. Then user can provide review for the books. Users can provide reward for every author.

4.2 FUNCTIONAL REQUIREMENTS

Login and Register- New users can register themselves by providing their details to log in. Once the user is registered in the application, then the user can be logged into the application and the user can provide a review of the book.

Update User – Only the registered users can have access to update their user's profile which consists of name, email, password, and genres after saving the details. Users can log in with their new details which are displayed in their profile.

Blog (Review for book) – After the reader reads the book, the reader will provide the review which helps the other users to refer to that and users can get books based on a review of that book.

Add Book – Once the author registers themselves. then the author can add books to the websites and also the list of books is displayed for the authors. The author will get rewards after the reader buys the books.

Recommendation - At the time of purchase, the user can get book recommendations by providing book title then the recommender will provide list of books as per the book title.

Searching of Books - The reader can search for the book after the user provides a book name and based on the book name result will be filtered, display the book name.

Subscription Plan - All users have to register themselves. The user has to choose a plan based on the feature provided by that plan user can get benefits which include download and purchase books.

Purchase of Book – During the time of purchase based on the subscription, plan users can get exclusive offers like price, download, etc.

Payments of Book – After the user selects the book, provide bank details and make the payment. These details will help the user to check cart information and verify the bill amount as per price.

Downloads of EBook – Users can have the option of purchase hard copies or download a soft copy of books with special discounts based on subscription plans.

4.3 NON FUNCTIONAL REQUIREMENTS

Availability - This system can be accessed anytime anywhere, with pertaining to internet availability.

Portability - This application in respect of different browsers in different devices it can be accessed.

Security - This application can access only by the registered users.

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 ARCHITECTURAL DIAGRAM

BOOKSEEKER

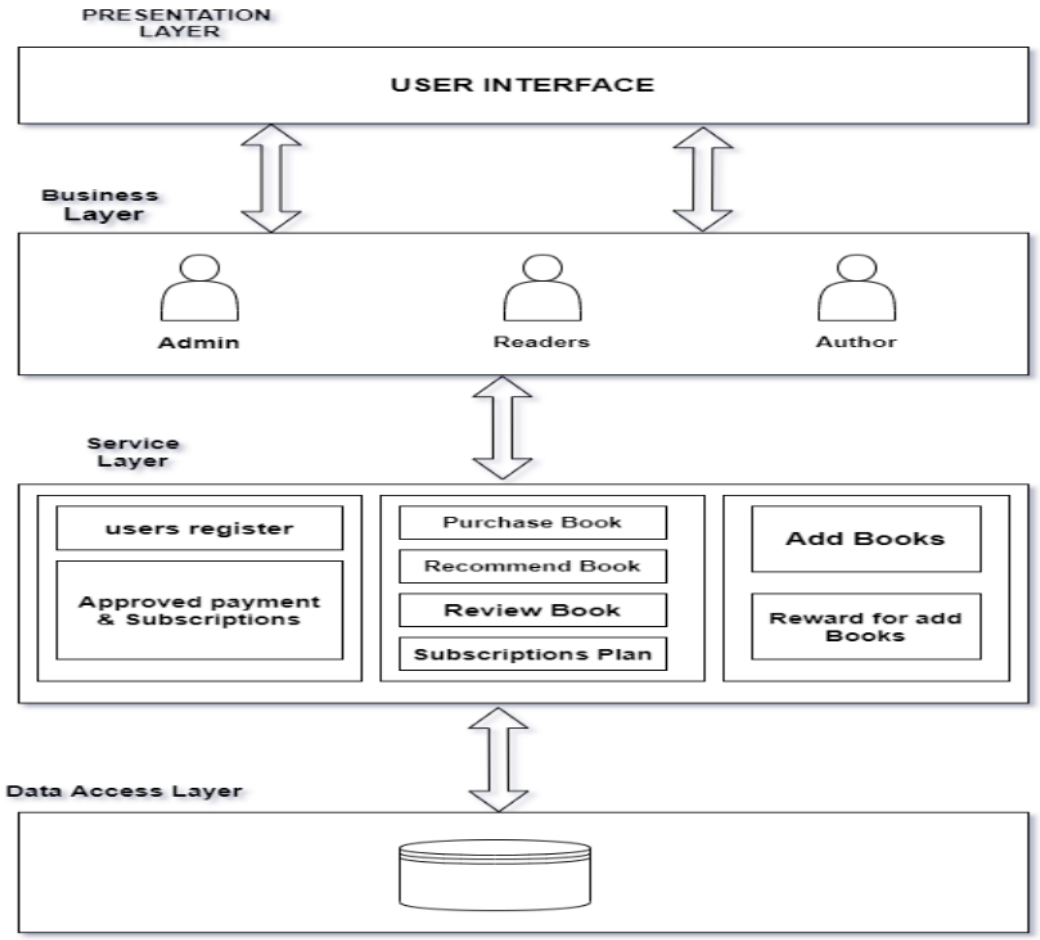


Fig 5.1 Architecture Diagram

Above figure 5.1 show how the user can interact with the system and how each layer interacts with each other. The presentation layer is just like User interface. Business layer is where logic is stored. Service layer is consisting of functionality of the system. Last is the database where data is store.

5.2 CONTEXT DIAGRAM

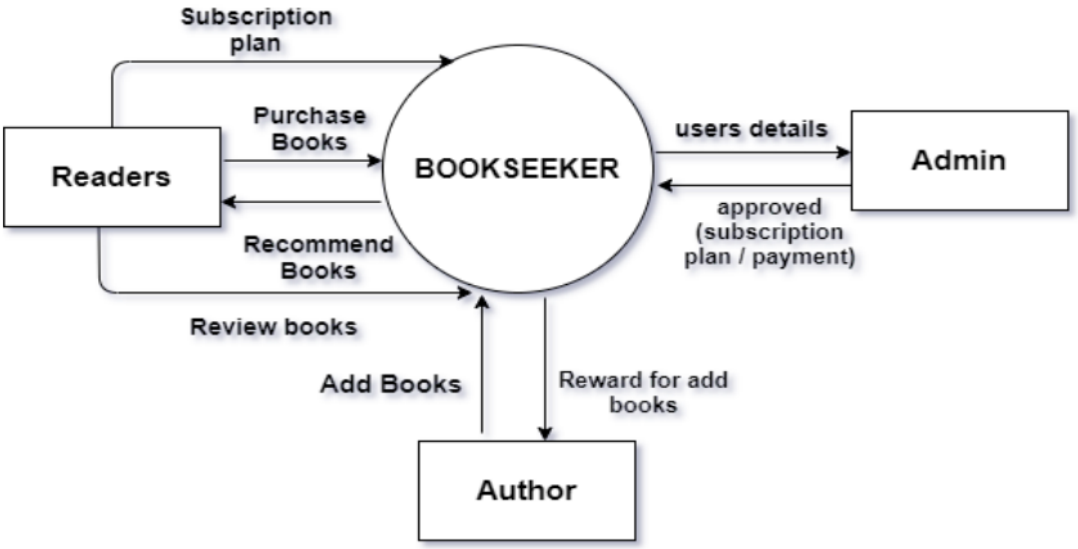


Fig 5.2 Context Diagram

The above figure 5.2 Data flow diagram show how the data flow through the system. Each user will get data based on the accessibility.

DETAILED DESIGN

6. DETAILED DESIGN

6.1 USER CASE DIAGRAM

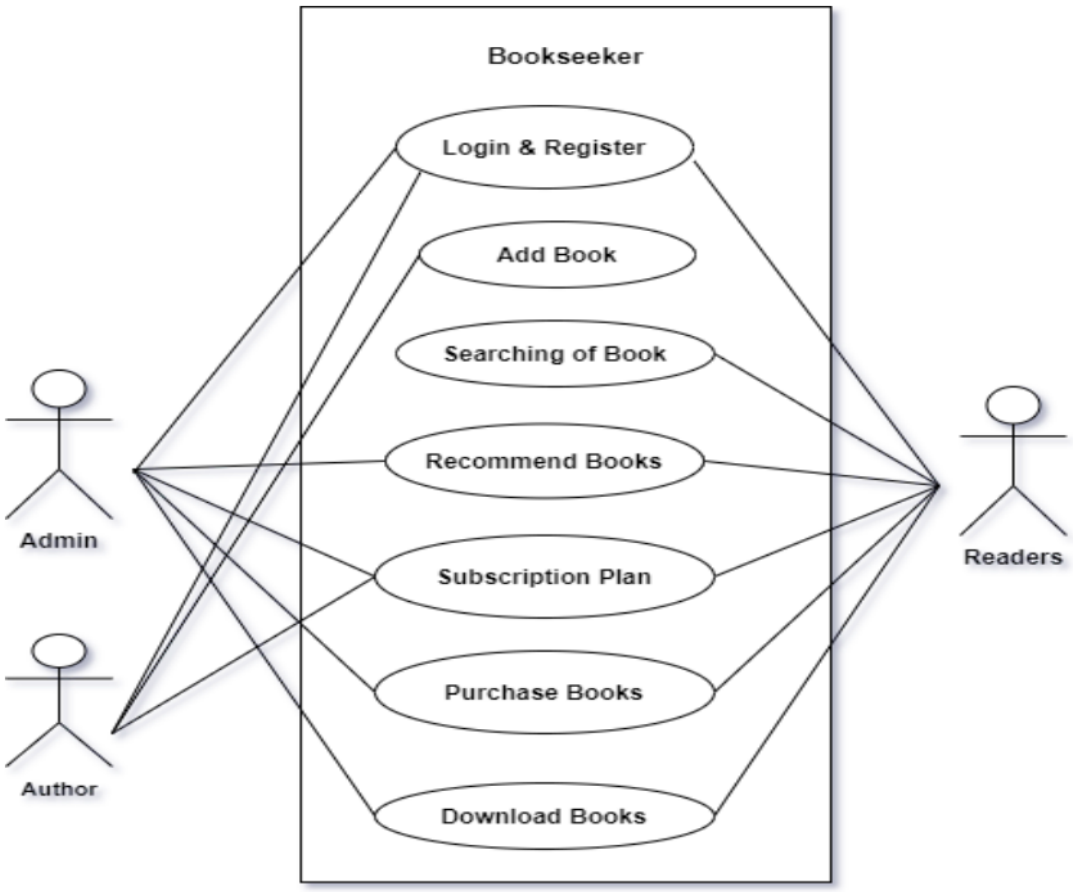


Fig 6.1 Use Case Diagram

The above figure 6.1 Use Case Diagram shows how the actors interact with the system.

Each actor need to login to interact with other actor, access the services.

Author can add book for users, and after purchase book the author will get credit.

Reader can purchase book, also provide review for books and reader can apply

subscription plan so that they can download book.

Admin can approve payment and subscription plan applied by the users.

6.2 ACTIVITY DIAGRAM

6.2.1 USER ACTIVITY DIAGRAM

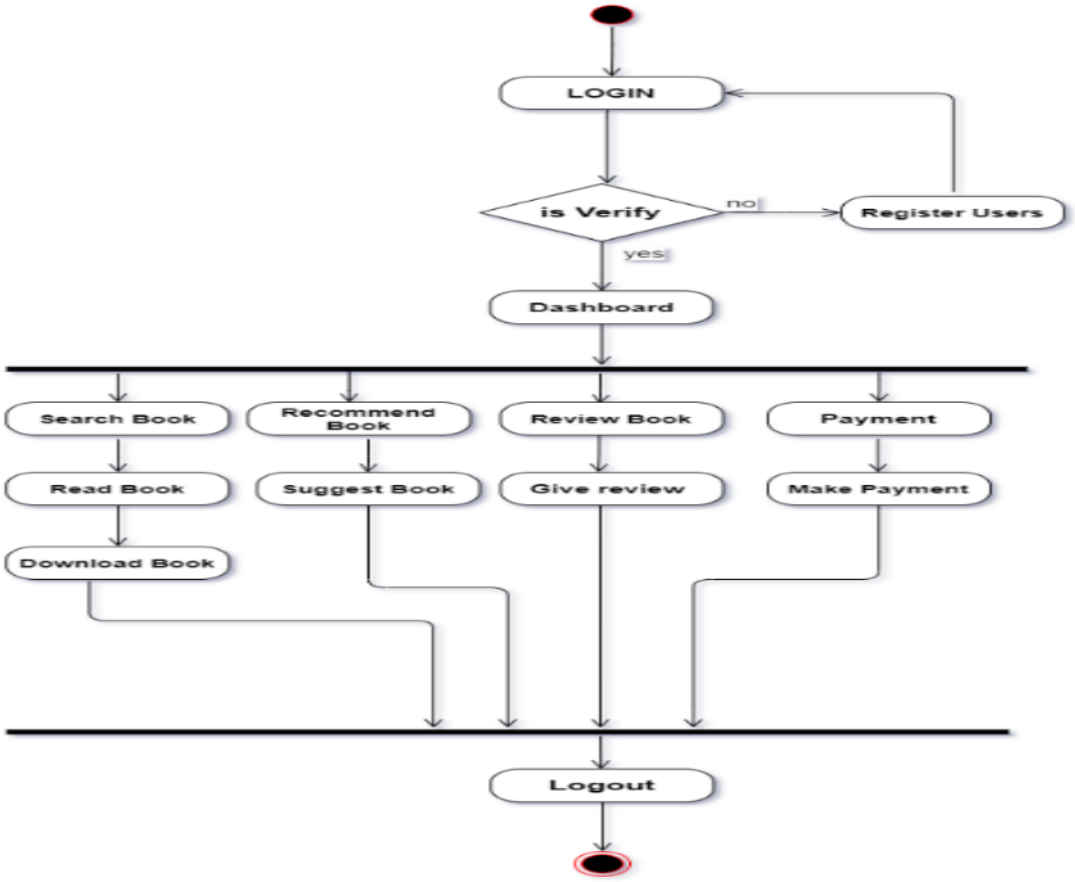


Fig 6.2.1 User Activity Diagram

6.2.2 ADMIN ACTIVITY DIAGRAM

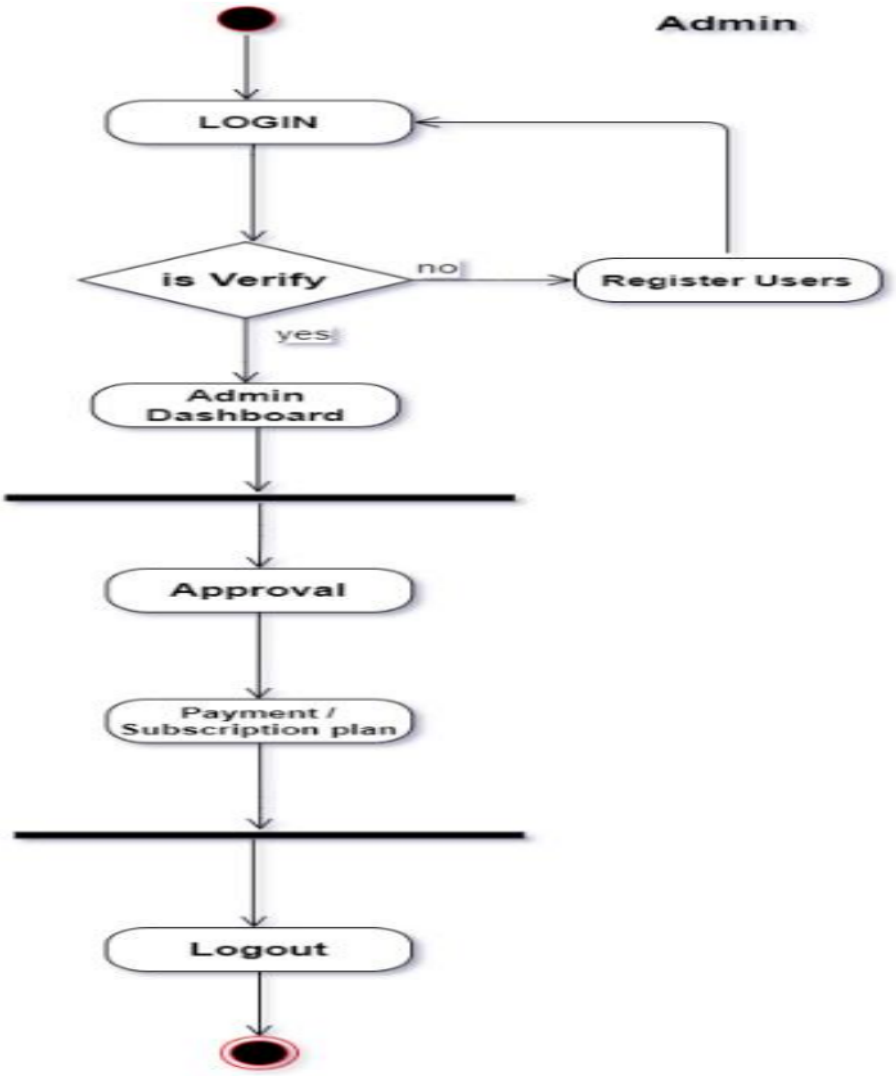


Fig 6.2.2 Admin Activity Diagram

6.2.3 AUTHOR ACTIVITY DIAGRAM

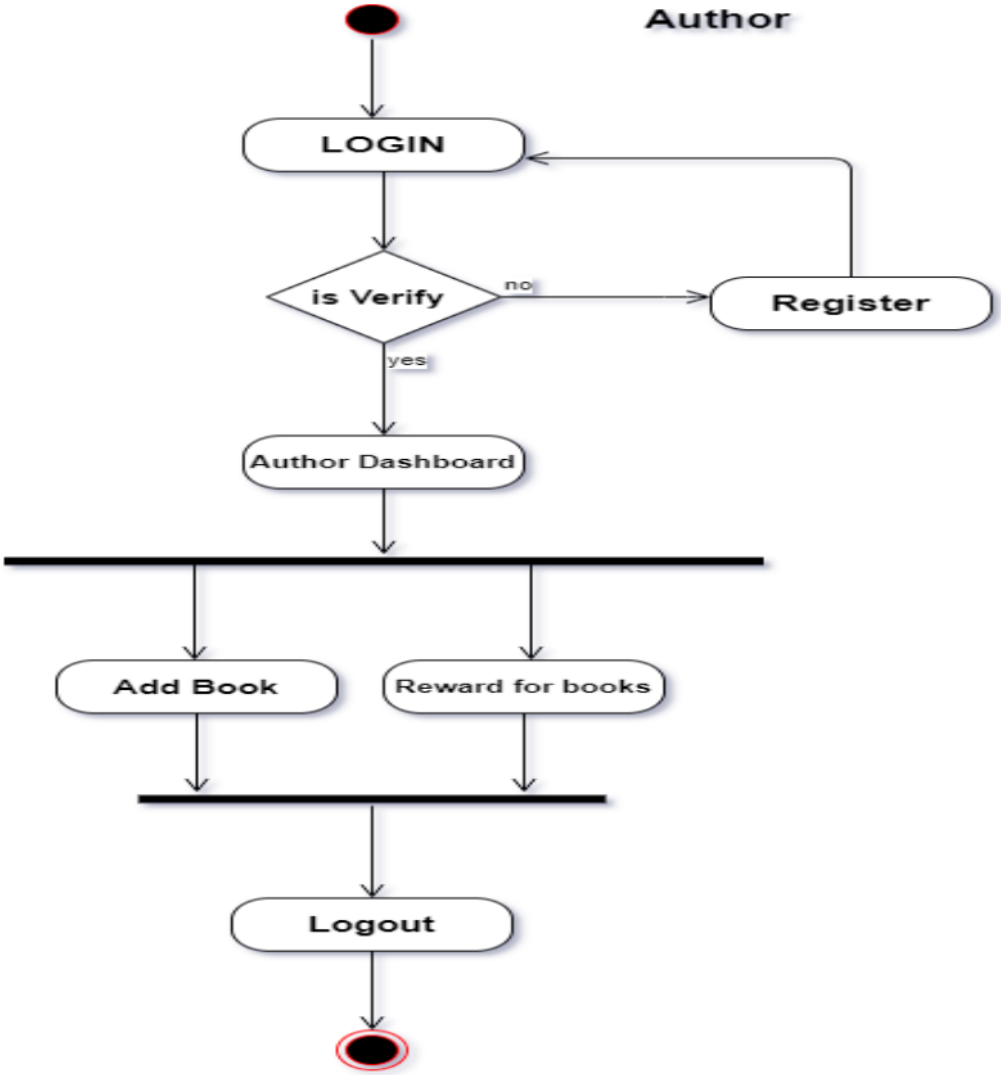


Fig 6.2.3 Author Activity Diagram

The Above three diagram figure 6.2.1 to 6.2.3 activity diagram. Figure 6.2.1 is for user in which first user login in the system then provide the review for books.

Then user can purchase books make payment.

User can also recommend books.

Figure 6.2.2 admin can approve payment and subscription plan.

Figure 6.2.3 Author can add books to the system and also get credit for purchase.

6.3 DOCUMENT STRUCTURE

6.1 USER SCHEMA

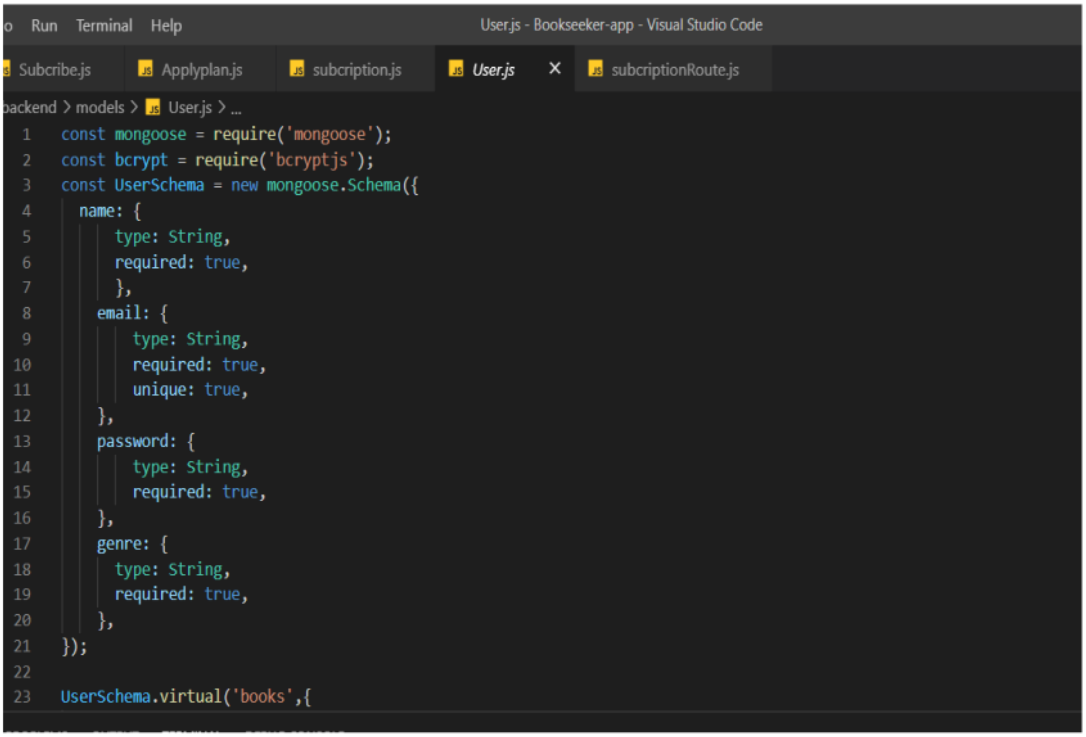


Fig 6.1 User Schema

6.2 BOOK SCHEMA

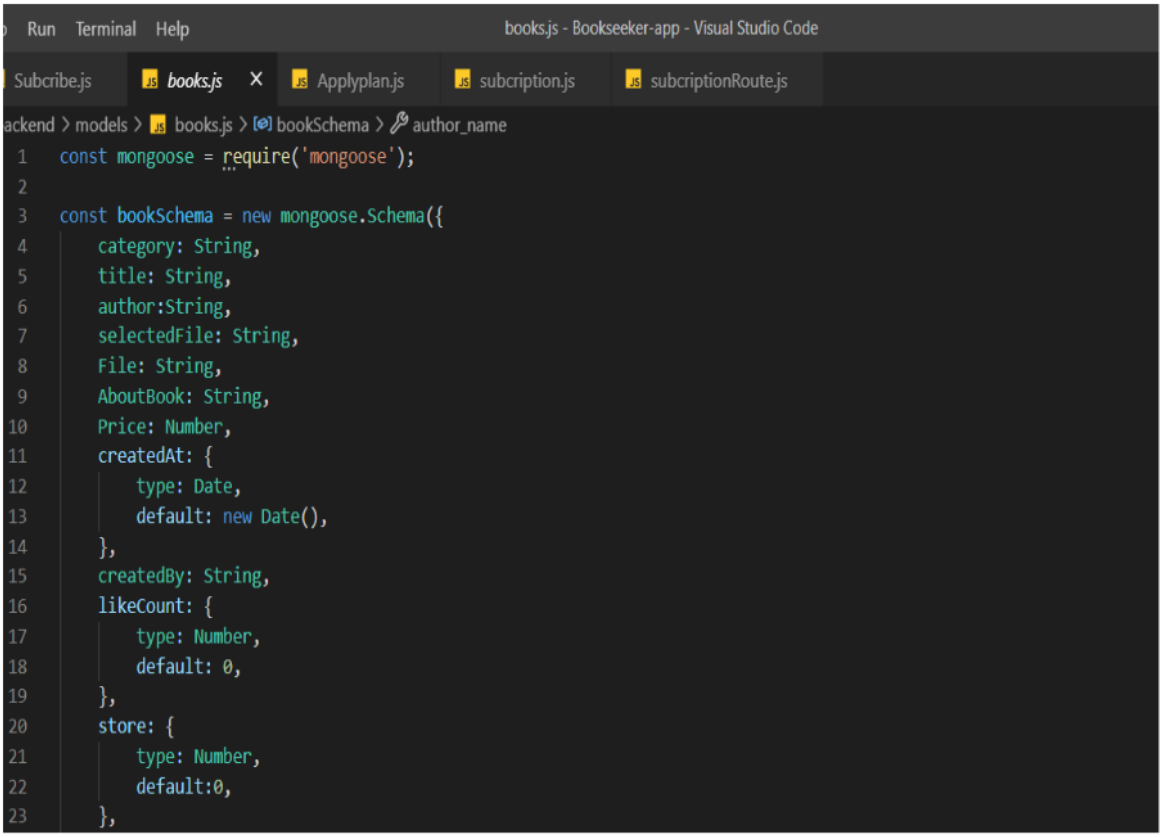


Fig 6.2 Book Schema

6.3 BLOG SCHEMA

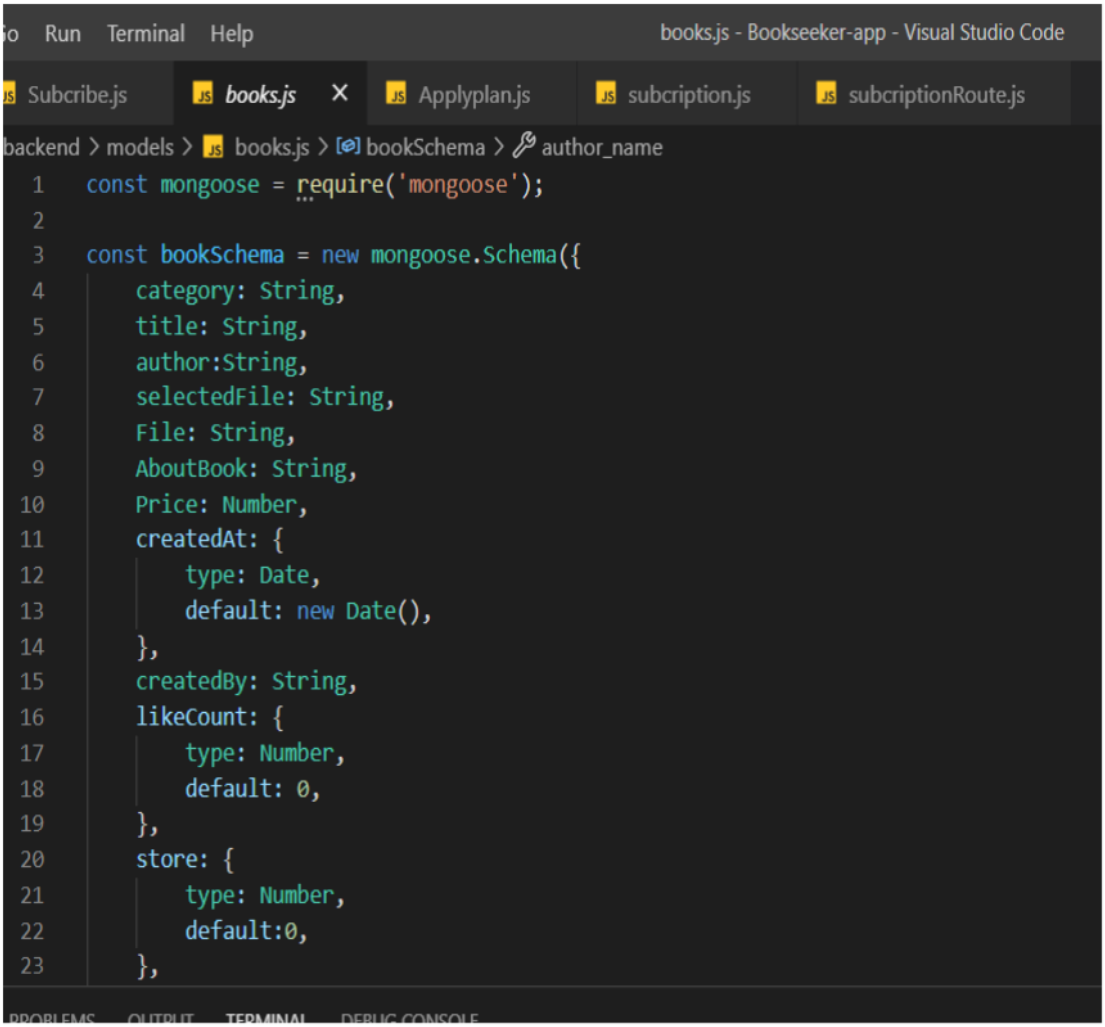


Fig 6.3 Blog Schema

6.4 CREDIT SCHEMA

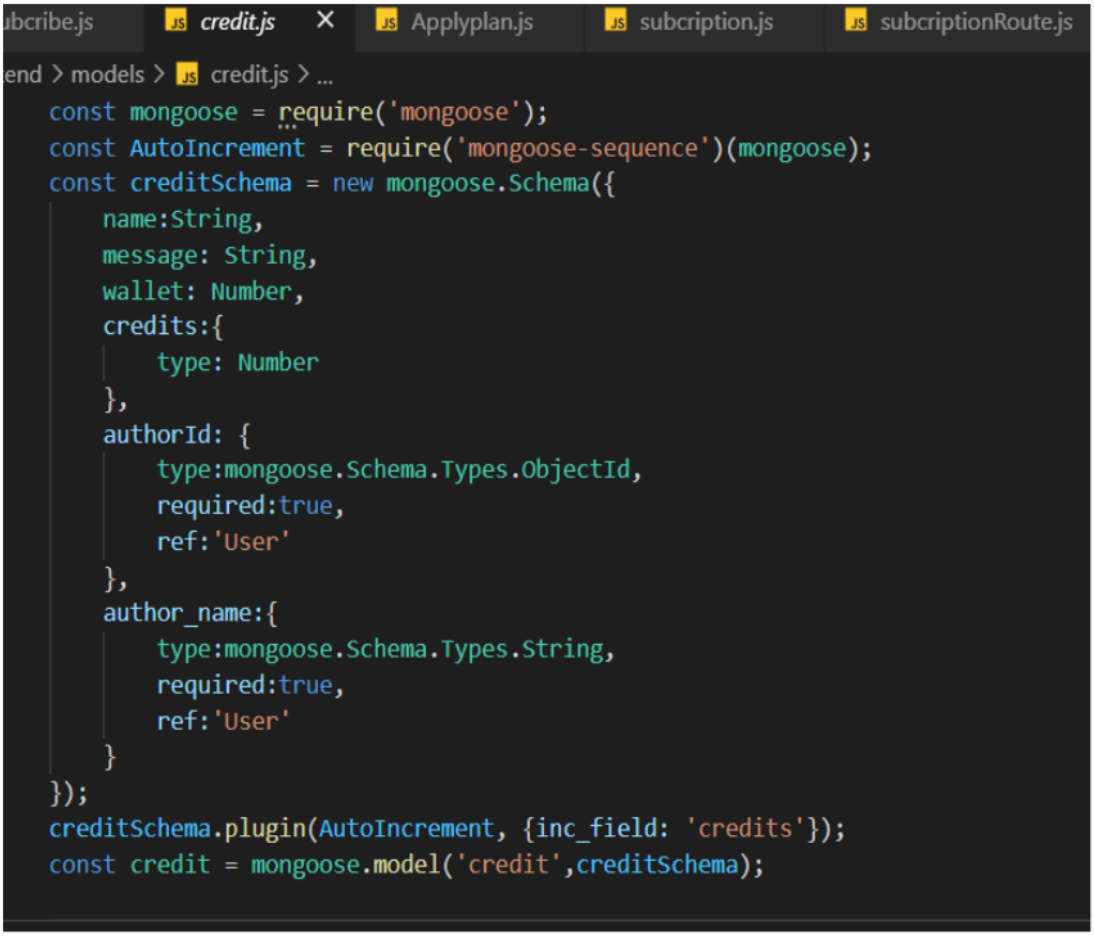


Fig 6.4 Credit Schema

6.5 DOWNLOAD SCHEMA

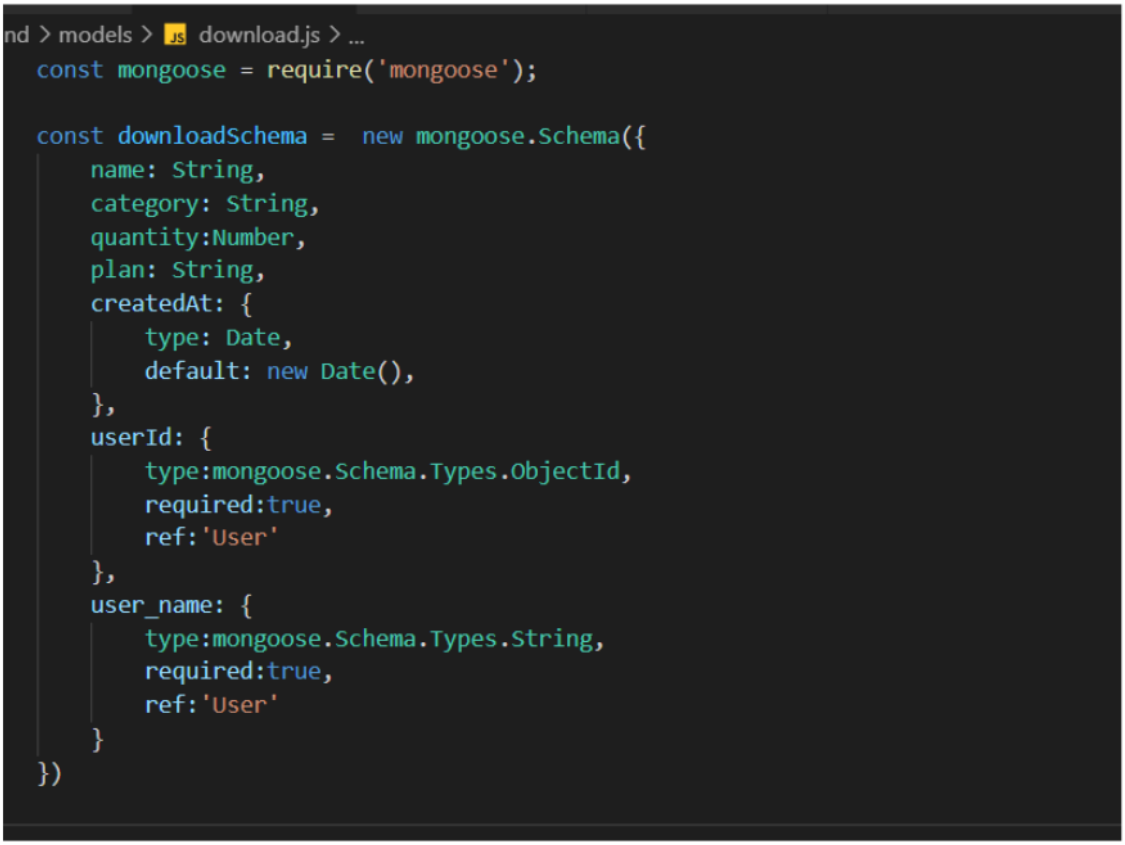


Fig 6.5 Download Schema

6.6 PAYMENT SCHEMA

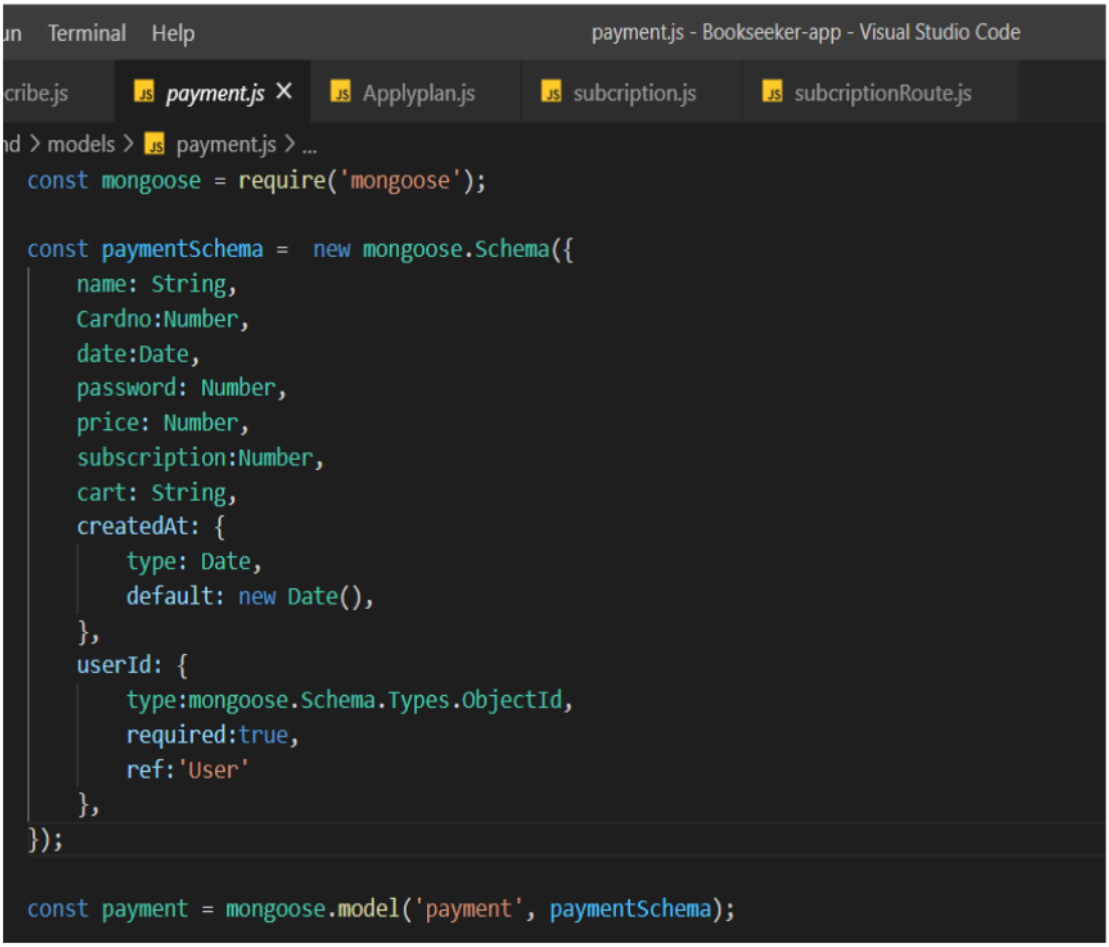
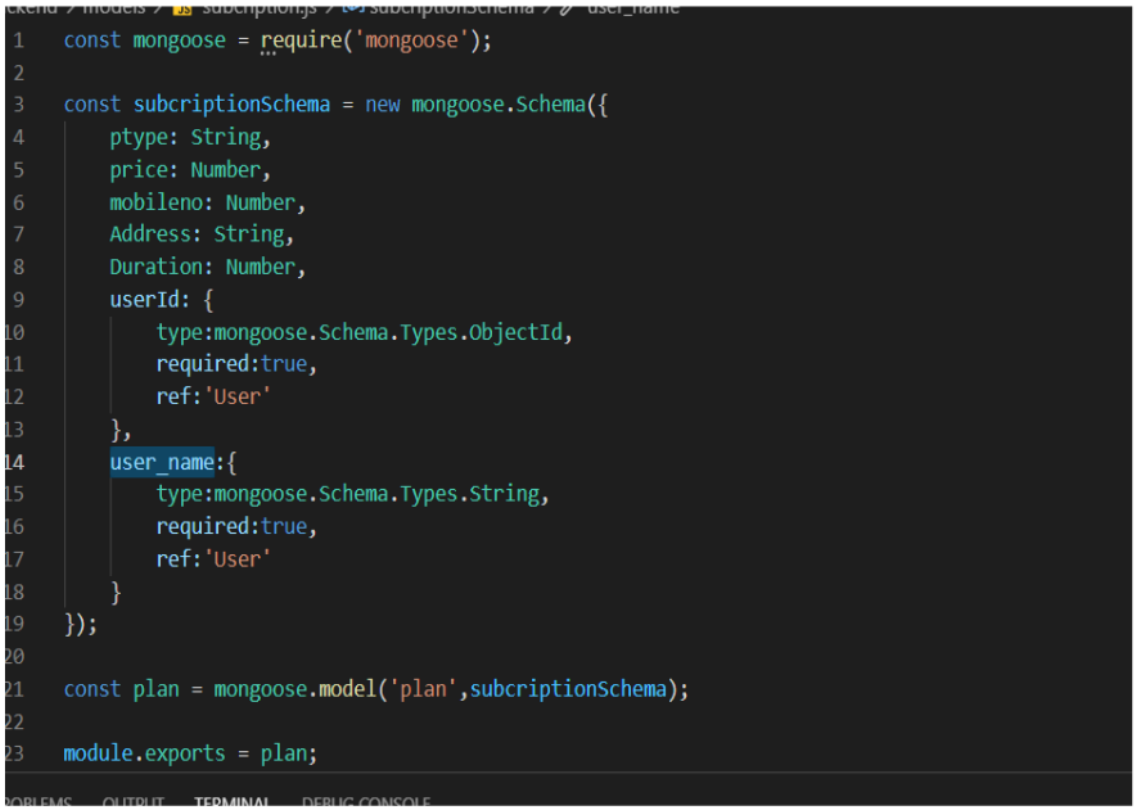


Fig 6.6 Payment Schema

6.7 SUBCRIPTION PLAN SCHEMA



```
const mongoose = require('mongoose');

const subcriptionSchema = new mongoose.Schema({
    ptype: String,
    price: Number,
    mobileno: Number,
    Address: String,
    Duration: Number,
    userId: {
        type:mongoose.Schema.Types.ObjectId,
        required:true,
        ref:'User'
    },
    user_name:{
        type:mongoose.Schema.Types.String,
        required:true,
        ref:'User'
    }
});

const plan = mongoose.model('plan',subcriptionSchema);

module.exports = plan;
```

Fig 6.7 Subscription plan Schema

IMPLEMENTATION

7. IMPLEMENTATION

7.1 PRODUCT VIEW

```
import React from 'react';
import './Product.css';
import { Link } from 'react-router-dom';
import { connect } from 'react-redux';
import { addToCart, loadCurrentItem } from
'../../../../Redux/actions/cart/cartAction';

const Product = ({productData, addToCart, loadCurrentItem}) => {
  return (
     <>
     <div className="container">
      <div>
         <img src={productData.imgurl} alt="" className="i1"/>
      </div>
      <div className="product_info">
       <p className="title_info">{productData.title}</p>
       <p className="desc_category">{productData.category}</p>
       <p className="desc_info">{productData.description}</p>
       <p className="price_info">Rs {productData.price}</p>
      </div>
      <div className="button">
          <button className="cart" onClick={() =>
addToCart(productData.id)}>Add To Cart</button><br/>
         <Link to={`/product/${productData.id}`}>
            <button className="view"  onClick={() =>
loadCurrentItem(productData)}>
                View Item
             </button>
         </Link>
      </div>
     </div>
     </>
   );
};
```

This above code is for product view which display list of products and this product are also stored in the shopping cart.

7.2 RECOMMENDATIONS VIEW

```
  import React, { useState } from 'react';
import './Recommender.css';


const Recommender = () => {
   return(
    <div className="recommends">
    <h1 className="recommends">Book Recommender</h1>
     <p className="results"><a
href="http://127.0.0.1:5000">Recommendations</a></p>
   </div>
   );
};


export default Recommender;
```

this above code is for recommending books based on book title and users have to provide book title than the user will get list of books.

7.3 SUBCRIPTION PLAN

```
import React, { useState, useEffect} from 'react';
import { useDispatch, useSelector } from 'react-redux';
import { createsubcription } from '../../Redux/actions/subcription/subcription';
import { Button, Paper, TextField } from '@material-ui/core';
const Applyplan = ({ currentId }) => {
   const [ subcriptiondata , setsubcriptiondata ] = useState({  ptype:'',price:'',
mobileno:'', Address:'',Duration:''});
   const subcribes = useSelector((state) => (currentId ? state.subcribes.find((ptype)
=> ptype._id === currentId): null));
   const dispatch = useDispatch();
   const pricelimit = 4;
   const planlimit = 20;
   const addresslimit = 300;
   const phone = 10;

   useEffect(() => {
    if (subcribes) setsubcriptiondata(subcribes);
   },[subcribes]);


 const onhandlesubmit = async (e) => {
   e.preventDefault();
   dispatch(createsubcription(subcriptiondata));
   Message();
 }

 const Message = () => {
  alert('subscribed is successfull');
 }
   return(
      <>
      <div style={{padding:"22px"}}>
      <h1 style={{fontFamily:"cursive",fontVariant:"all-petite-
caps"}}>Subscription Form </h1>
      </div>
      <div>
```

```
          <form style={{width:"460px",height:"490px",
paddingLeft:"575px",marginLeft:"98px"}} onSubmit={onhandlesubmit}>

<Paper style={{width:"365px",height:"470px", padding:"23px"}}>
            <TextField type="text"  placeholder="plan type"
value={subcriptiondata.ptype}  onChange={(e) => setsubcriptiondata({
...subcriptiondata, ptype:e.target
.value})} inputProps={{maxlength:planlimit,pattern:"[A-Za-z ]*$"}}
helperText="choose the plan type" required/><br/><br/>
            <TextField type="text"  placeholder="mobile number"
value={subcriptiondata.mobileno}  onChange={(e) => setsubcriptiondata({
...subcriptiondata, mobileno:e.target.value})}
inputProps={{maxlength:phone,pattern:"^(0|91|\+91)?-?[789]\d{9}$"}}
helperText="provide the mobile number" required/><br/><br/>
            <TextField type="text"  placeholder="Address"
value={subcriptiondata.Address}  onChange={(e) => setsubcriptiondata({
...subcriptiondata, Address:e.target.value})}
inputProps={{maxlength:addresslimit,pattern:"[A-Za-z ]*$"}}
helperText="provide the address" required/><br/><br/>
            <TextField type="text"  placeholder="plan price"
value={subcriptiondata.price}  onChange={(e) => setsubcriptiondata({
...subcriptiondata, price:e.target.value})}
inputProps={{maxlength:pricelimit,pattern:"^[0-9]+$"}} helperText="provide the
price for plans" required/><br/><br/>
            <TextField type="text"  placeholder="plan duration"
value={subcriptiondata.Duration}  onChange={(e) => setsubcriptiondata({
...subcriptiondata, Duration:e.target.value})} inputProps={{pattern:"^[0-9]+$"}}
helperText="provide the duration for plans" required/><br/><br/>
            <Button variant="contained" color="primary"  size="auto" type="select"
fullwidth onDBClick={Message}><b>Submit</b></Button>
          </Paper>
          </form>
       </div>
       </>
   );
};
```

This above code is for subscribing plan for every user but only for register users. The user will exclusive offer after the user have subscribe the plan like download of softcopy etc.

7.4 LOGIN VIEW

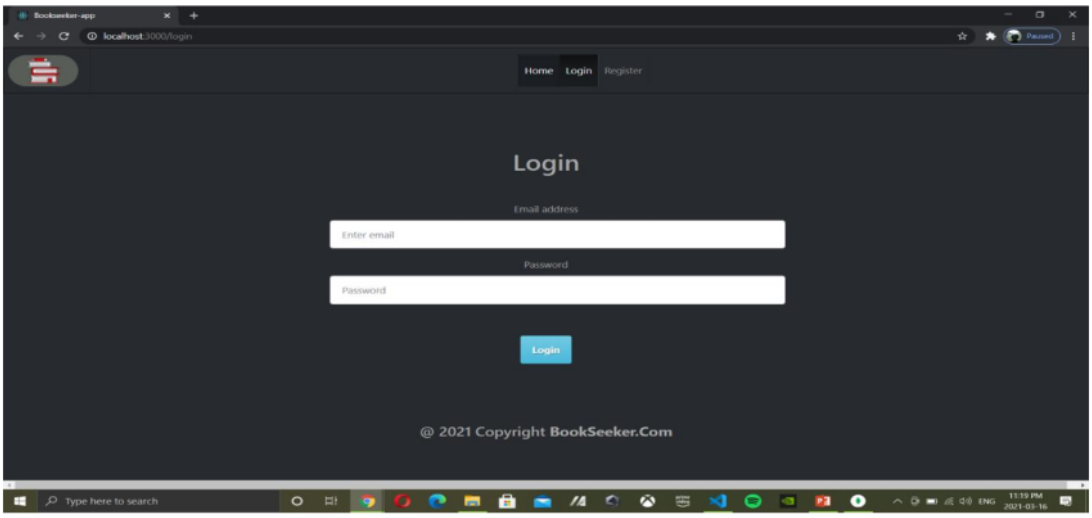


Fig 7.4 Login View

7.5 REVIEW VIEW

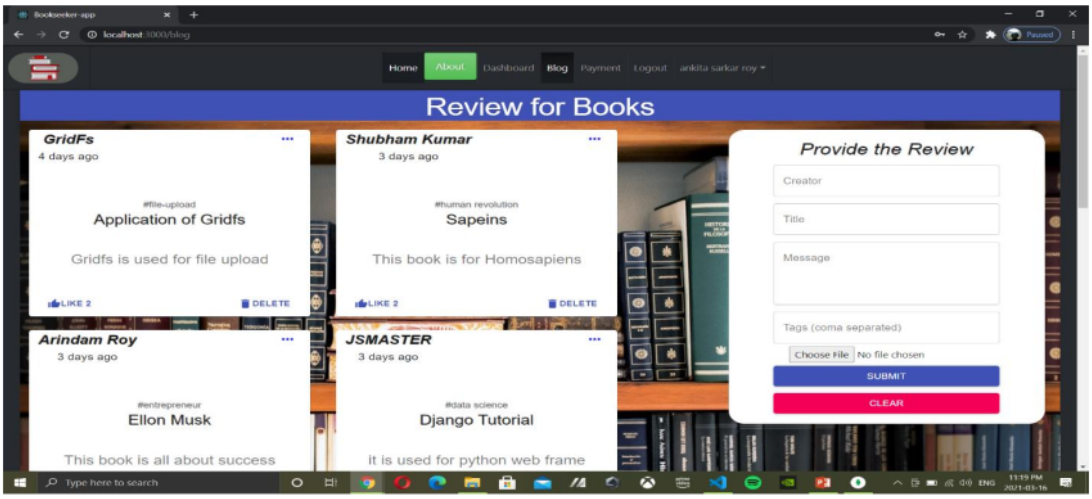


Fig 7.5 Review View

7.6 PROFILE VIEW

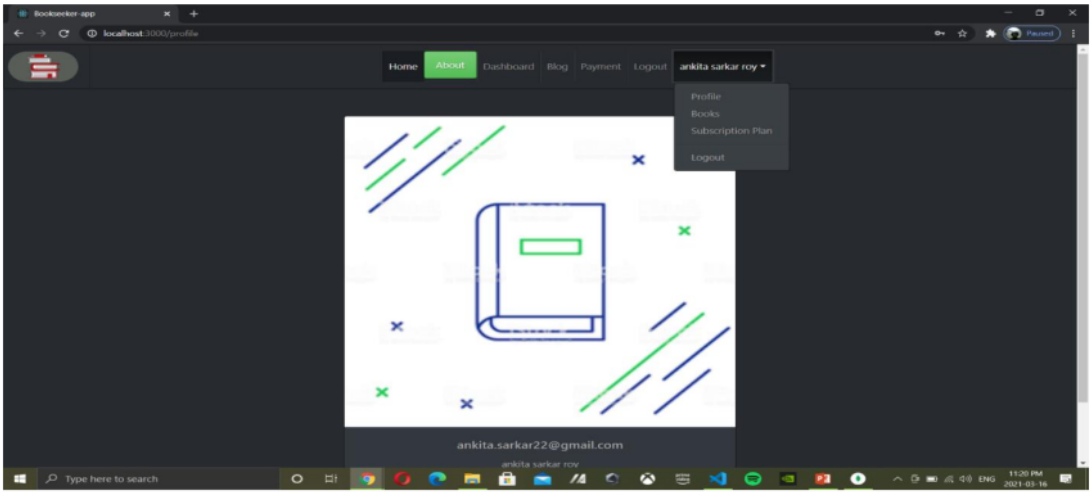


Fig 7.6 Profile View

7.7 CATALOG VIEW

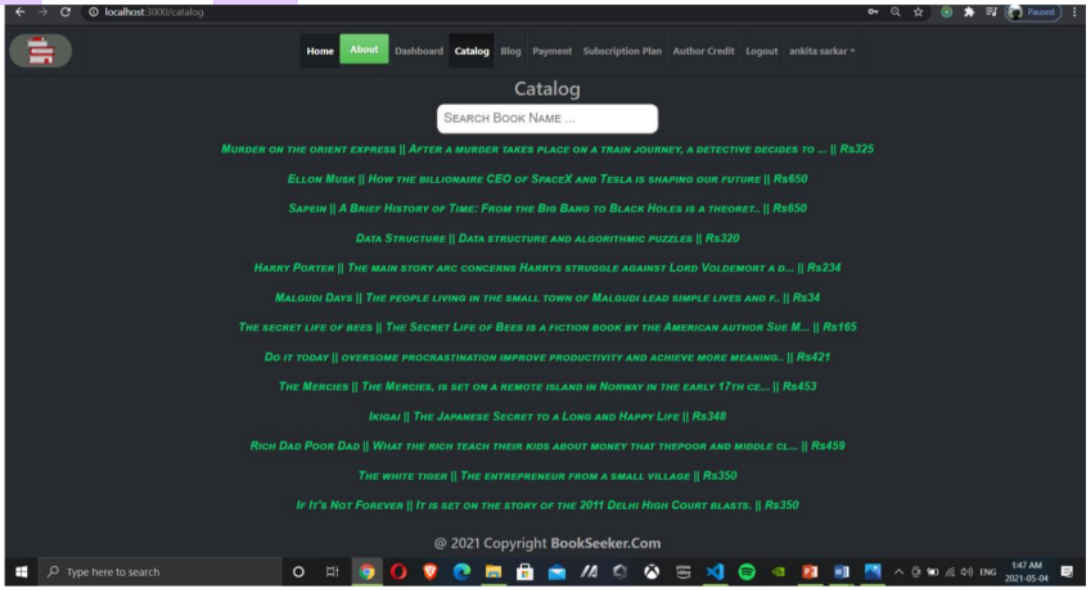


Fig 7.7 Catalog view

7.8 ADD BOOK VIEW

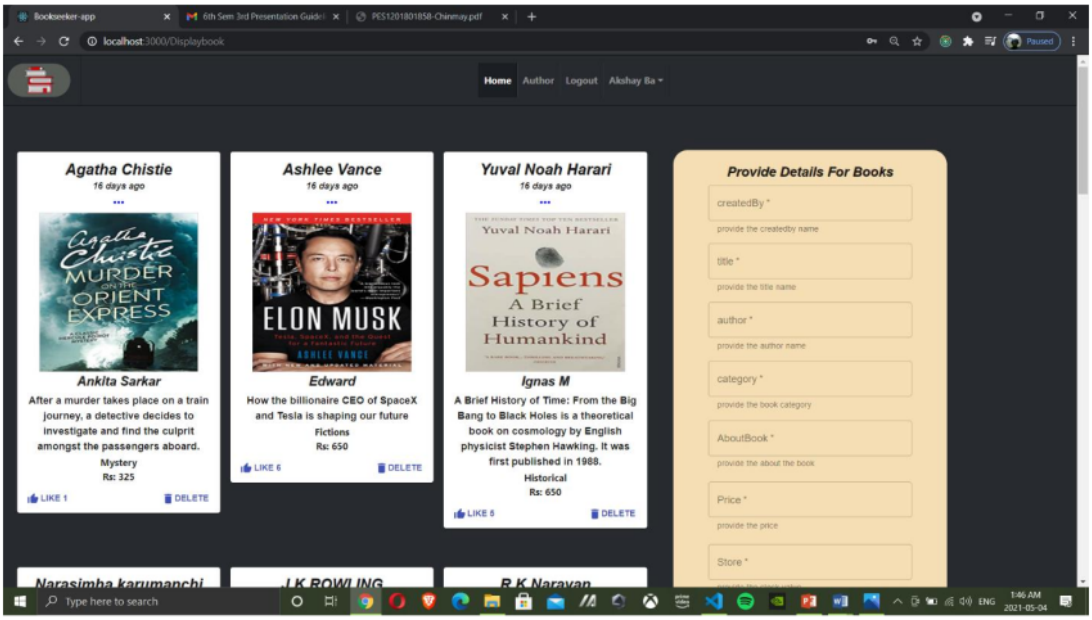


Fig 7.8 Add Book View

7.9 SHOPPING CART VIEW

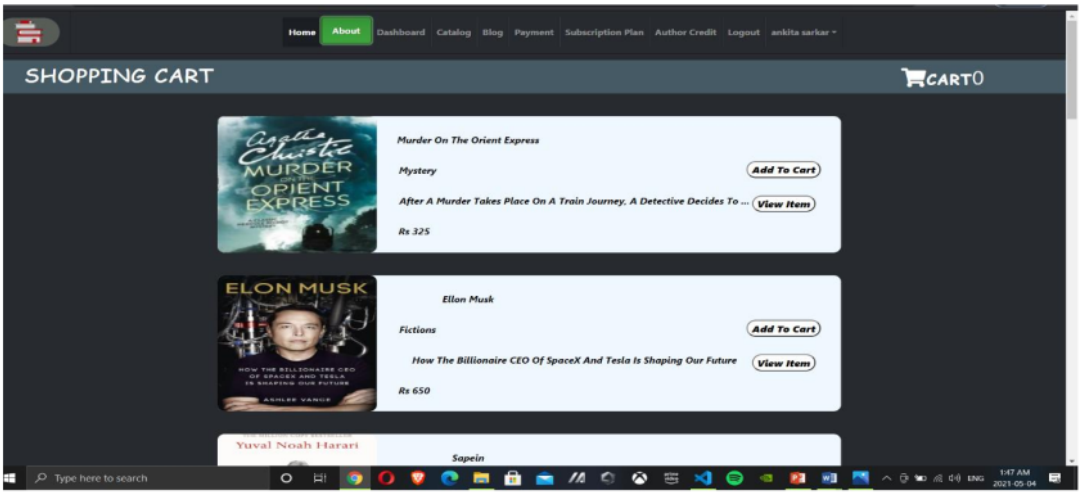


Fig 7.9 Shopping cart view

7.10. DOWNLOAD VIEW

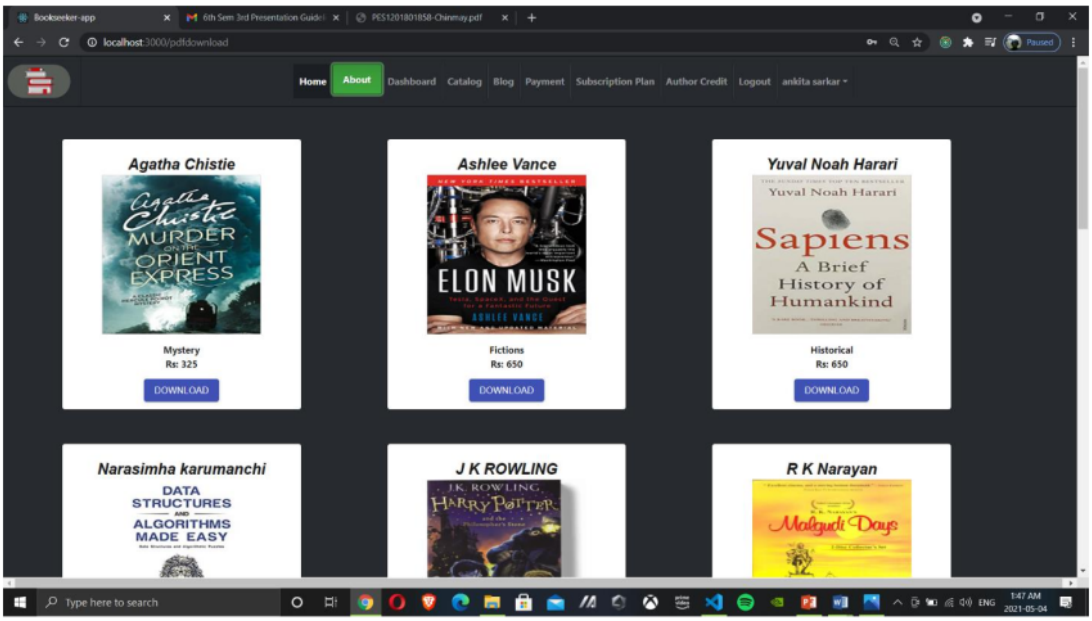


Fig 7.10 Download view

7.11 SUBSCRIPTION PLAN VIEW

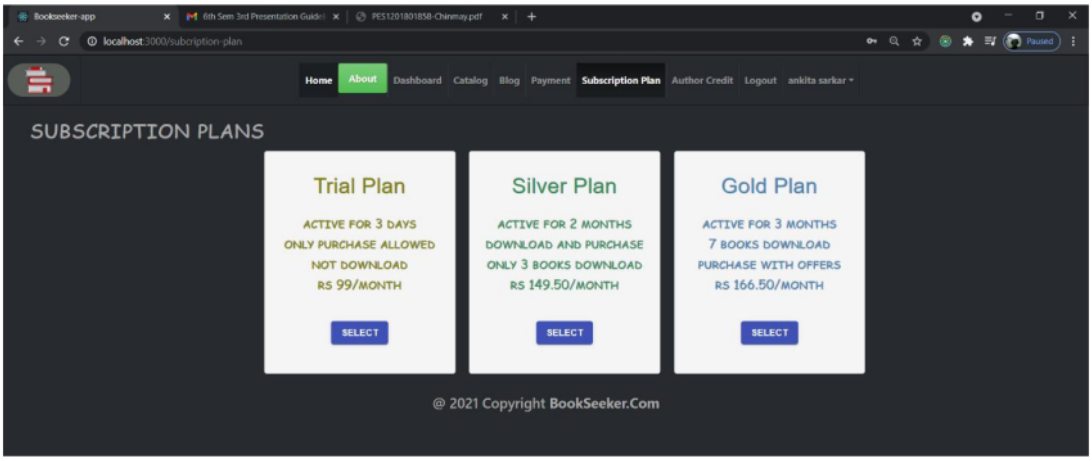


Fig 7.11 Subscription plan

7.12 ADMIN DASHBOARD VIEW

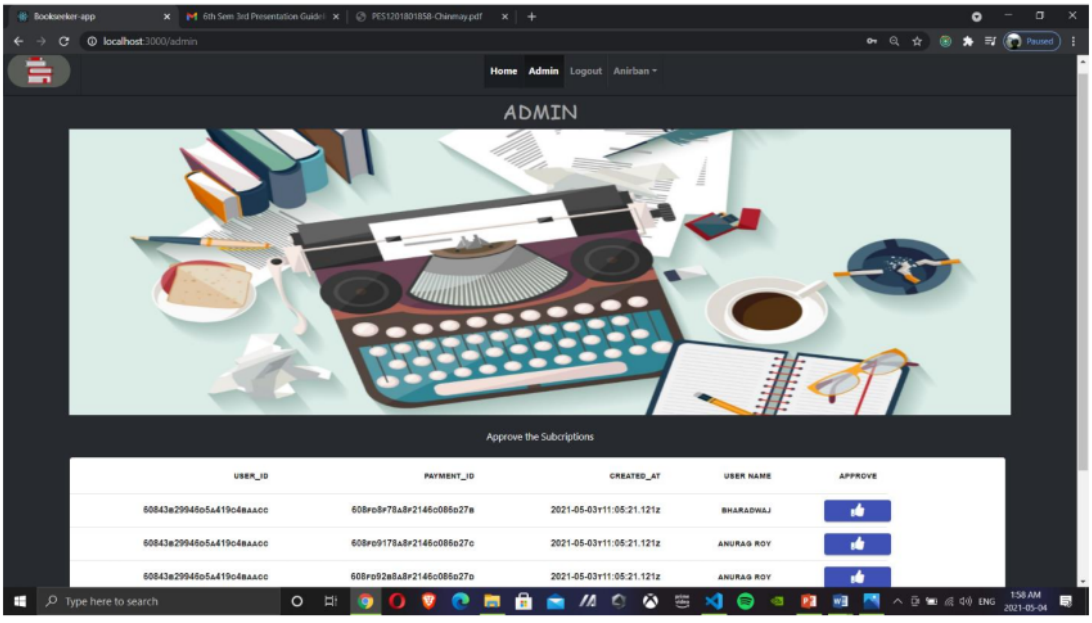


Fig 7.12 Admin view

7.13 AUTHOR DASHBOARD VIEW

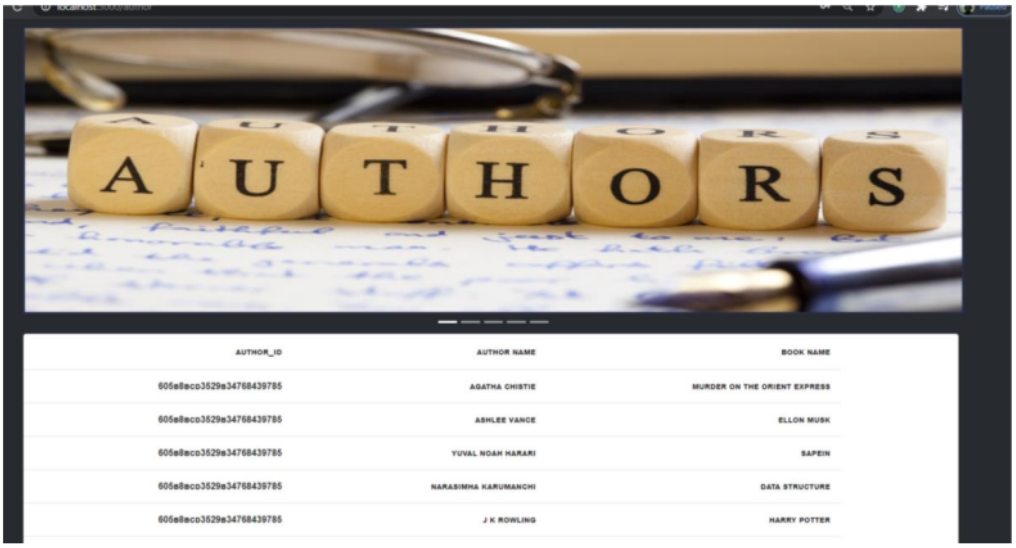


Fig 7.13 Author view

7.14 PAYMENT VIEW

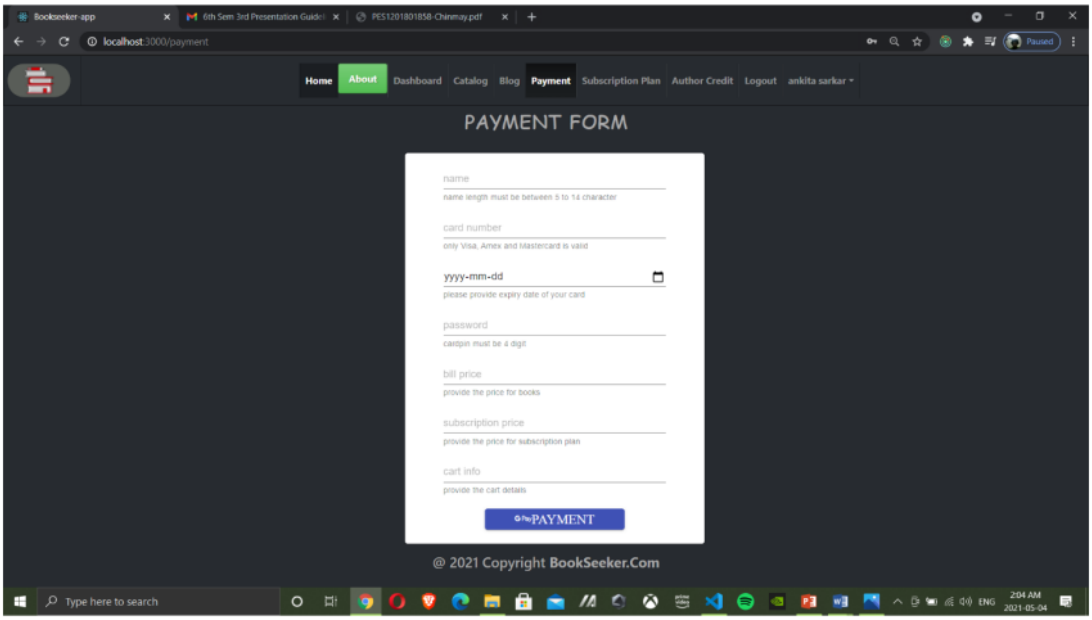


Fig 7.14 Payment view

7.15 AUTHOR CREDIT VIEW

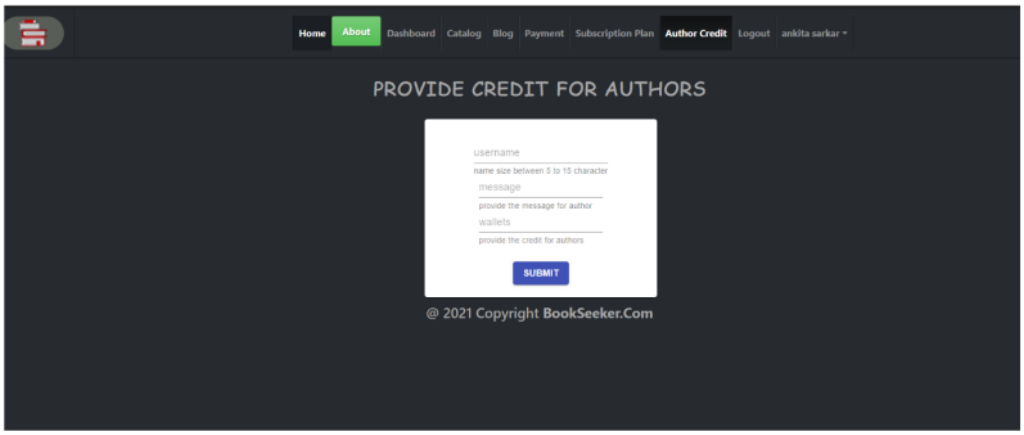


Fig 7.15 Author credit

7.16 Apply plan view

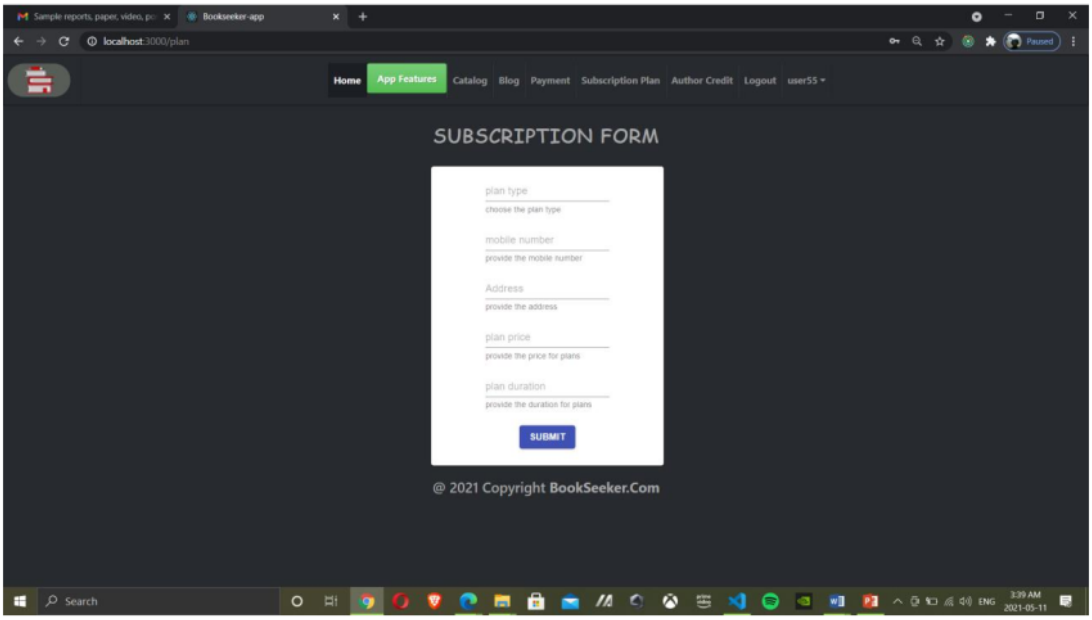


Fig 7.16 Apply plan view

SOFTWARE TESTING

8. SOFTWARE TESTING

8.1 AUTHOR AND USER TEST CASES

Test ID	Description	Input	Expected Result	Actual Result	Test Result
1	Registration	User Details	Details will be validated and stored in the database	User will be redirected to homepage	PASS
			If user email is already registered	User will be redirected to homepage	PASS
2	Authentication	Valid email and password	User need to redirected to the dashboard	User will be redirected to homepage	PASS
		Invalid email and password	User need to be login again	User need to login again	PASS

3	Subscription	Valid plan details	If the user filled the plan details.	The plan details will be stored in the database	PASS
4	Blog	Valid blog details	If the user filled the blog details	The blog info will be stored in the database	PASS
5	Add Book	Valid book details	If the user filled the book details	The book info will be stored in the database	PASS
6	Payment	Valid payment details	If the user filled the payment details	The payment info will be stored in the database	PASS
7	Download	Valid download details	If the user filled the download details	The download info will be stored in the database	PASS

8	Credit	Valid credit details	If the user filled the credit details	The credit info will be stored in the database	PASS
9	Recommendations	Valid recommend	Display recommend books	Recommend books	PASS
10	Profile	Valid Profile info	If the user filled the profile details	Profile info will be updated	PASS

Fig 8.1 User test

8.2 Admin Test case

Test ID	Description	Input	Expected Result	Actual Result	Test Result
1	Login	Valid email and password	Admin will be redirected to the admin dashboard	Admin will be Redirected to the homepage	PASS
2	Approve Payment	Payment is done	Payment_Id is remove the table	Payment is done	PASS
3	Approve Subscription	Subscription is done	Subscription_Id is remove the table	Subscription is done	PASS

Fig 8.2 Admin test

CONCLUSION

9. CONCLUSION

The Bookseeker is a unified platform is created that assists book lovers at, purchase a paper copy of the books in accordance with the user's choice. Basically, this application allows the user to select the book in accordance with the choice of the user, same books can be downloaded in pdf format. This application helps to save time on. book enthusiasts search book from catalog, book recommendation is possible. Based on the information in the books. The app also makes it possible for the user to provide. Subscriptions to registered users and attractive offers on the purchase of books.

FUTURE ENHANCEMENTS

10. FUTURE ENHANCEMENTS

- User can share books through online.
- More flexibility in subscription plan.
- Recommendation using collaboration filtering.

APPENDIX

Appendix A: BIBLOGRAPY

Books References

- The Road to React Book by Robin Wieruch

Web References

- https://redux.js.org/tutorials/fundamentals/part-1-overview
- https://reactjs.org/tutorial/tutorial.html
- https://www.tutorialspoint.com/mongodb/mongodb_relationships.htm
- https://www.amazon.in/
- https://www.amazon.in/Kindle-Store/b/?ie=UTF8&node=1571277031&ref_=nav_cs_kindle_books_fbe4f5e5a26f403a91203de3fe5ab05d
- https://www.librarything.com/
- https://www.goodreads.com/?shelfari_flash=true
- https://www.riffle.com/

Appendix B: USER MANUAL

AUTHOR

- The first author must register its account.
- Go to the log in page to log in as an author.
- Now, after logging on, the author may add a book to the system.
- Author can also view the list of books.

USER(READER)

- At the starting user have to register to the system then user can logged into the system.
- The reader is also able to provide a review after reading the books.
- For purchasing book user have to navigate to shopping cart select the item add to cart and select the quantity then go to checkout page.
- After adding item to the cart user can go for payment.
- For downloading book user has to apply subscription plan based on the offer user can download books.
- After user purchase books credits will sent to authors.

BOOKSEEKER

ORIGINALITY REPORT

14%	11%	3%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to PES University Student Paper	3%
2	eprints.utem.edu.my Internet Source	1%
3	Dong-Chan Lee. "Protaspides of uppermost Cambrian trilobite *Missisquoia*, with implications for suprafamilial level classification of the genus", Canadian Journal of Earth Sciences, 04/2007 Publication	1%
4	ethesis.nitrkl.ac.in Internet Source	1%
5	en.wikipedia.org Internet Source	1%
6	Submitted to University of Brighton Student Paper	1%
7	Submitted to Higher Education Commission Pakistan Student Paper	1%

8	www.coursehero.com Internet Source	1 %
9	Submitted to The University of Manchester Student Paper	1 %
10	www.webfx.com Internet Source	1 %
11	Submitted to CSU, Fullerton Student Paper	1 %
12	Submitted to University of Bath Student Paper	<1 %
13	pt.scribd.com Internet Source	<1 %
14	Submitted to Aston University Student Paper	<1 %
15	chaplin.expertlead.com Internet Source	<1 %
16	Submitted to CSU, Los Angeles Student Paper	<1 %
17	ukdiss.com Internet Source	<1 %
18	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1 %
19	docplayer.net	

	Internet Source	<1%
20	www.cubix.com Internet Source	<1%
21	Submitted to CSU, San Jose State University Student Paper	<1%
22	Submitted to University of Greenwich Student Paper	<1%
23	Submitted to University of Westminster Student Paper	<1%
24	www.scribd.com Internet Source	<1%
25	K. Sudhakar .. "A NOVEL METHODOLOGY FOR DIAGNOSING THE HEART DISEASE USING FUZZY DATABASE", International Journal of Research in Engineering and Technology, 2015 Publication	<1%
26	Submitted to Sri Lanka Institute of Information Technology Student Paper	<1%
27	Submitted to Rajarambapu Institute of Technology Student Paper	<1%
28	www.antiessays.com Internet Source	<1%

29	Simone Chiaretta. "Front-end Development with ASP.NET Core, Angular, and Bootstrap", Wiley, 2018 Publication	<1 %
30	utpedia.utp.edu.my Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		