

Table of Contents

Chapter 1: Introduction	5
1.1. Terminology	5
1.2. Purpose.....	6
1.3. Motivation.....	6
1.4. Problem Statement.....	6
1.5. Objective.....	7
1.6. Scope and Relevance	7
Chapter 2: Requirement Specifications.....	8
2.1. Specifications	8
2.1.1. Functional Requirements.....	8
2.1.2. Non-Functional Requirements	9
2.1.3. User Interface Requirements	10
2.2. Hardware Requirements	11
2.3. Software Requirements	11
Chapter 3: Design.....	12
3.1. ER Diagram	12
3.1.1. E-R Diagram Representation	12
3.1.2. Schema Representation	13
3.2. UML Diagrams.....	14
3.2.1. Class Diagram	14
3.2.2. Sequence Diagram.....	15
3.2.3. Use Case Diagram.....	17
3.2.4. Activity Diagram.....	20
3.3. Front End Design.....	22
Chapter 4: Implementation Details.....	26
4.1. Database Implementation	26
4.1.1. Table Creation	26
4.1.2. Table Population	30
4.1.3. Query Execution and Output.....	32

4.2. Front End Implementation.....	34
4.2.1. Form Creation	34
4.2.2. Connectivity to Database	40
4.2.3. Report Generation	41
4.2.4. Security Features	42
Chapter 5: Testing and Results.....	43
5.1. Database Testing.....	43
5.1.1. Test Cases.....	43
5.2. Front End Testing	44
5.2.1. Test Cases.....	45
5.3. System Testing	46
5.3.1. Test Cases.....	46
Chapter 6: Conclusion.....	48
6.1. Limitations.....	48
6.2. Future Enhancements	48
References	49
APPENDIX A: Code Snippets	50
APPENDIX B: Screenshots	56

List of Figures

Figure 3.1: ER Representation for Online Platform	12
Figure 3.2: Schema Representation for Online Platform	13
Figure 3.3: Class Diagram for Online Platform	14
Figure 3.4: Sequence Diagram for Login Activity	15
Figure 3.5: Sequence Diagram for Purchasing Activity.....	16
Figure 3.6: Use Case diagram for purchasing a book.....	17
Figure 3.7: Use Case diagram for Shopping Cart.....	18
Figure 3.8: Use Case diagram for maintaining account	19
Figure 3.9: Activity Diagram for Admin View	20
Figure 3.10: Activity diagram for User View	21
Figure 3.11: Home Page Front End Design.....	22
Figure 3.12: Book Details Front End Design	22
Figure 3.13: Review and User Ratings Front End.....	23
Figure 3.14: Smart search using keywords Front End.....	23
Figure 3.15: Login Page	24
Figure 3.16: Sign up Form.....	24
Figure 3.17: Admin Dashboard	25
Figure 3.18: Added to Bookmark.....	25
Figure 4.1: Show Book with price less than Rs. 250.....	32
Figure 4.2: Show books from user named 'Harsh'	32
Figure 4.3: List Contact Numbers which begin with number 9	33
Figure 4.4: List all users under category 'Computer'.....	33
Figure 4.5: List all available categories.....	33
Figure 4.6: Form for Adding Book.....	35
Figure 4.7: Form for adding reviews	36
Figure 4.8: Form for transaction portal	36
Figure 4.9: Form for Editing User Details.....	38
Figure 4.10: Form for registering new user.....	39

Figure 4.11: Form for changing password	40
Figure 4.12: Connecting to database	40
Figure 4.13: Sample report generation to view all reviews.....	41
Figure 4.14: Report generation for Transactions.....	42
Figure 0.1: Code Snippet-1 "index.php" for homepage	53
Figure 0.2: Code Snippet - "search.php" for homepage	55
Figure 0.1: Homepage	56
Figure 0.2: Login Form	56
Figure 0.3: User Registration Form.....	57
Figure 0.4: View Book Details	57
Figure 0.5: Review Page.....	58
Figure 0.6: Form for adding a review	58
Figure 0.7: Form for Selling a Book	59
Figure 0.8: Admin Homepage	59
Figure 0.9: Admin view of All Reviews	60
Figure 0.10: Admin View of All Books	60

Chapter 1: Introduction

1.1. Terminology

The terminologies used in the following document are given below:

- **HTML** – Hypertext Markup Language is the standard markup language for creating web applications.
- **CSS** – Cascading Style Sheets is the most widely used styling and presenting the webpages designed using a markup language like HTML.
- **PHP** – Hypertext Preprocessor is a general purpose programming language originally designed for web technologies.
- **JS** – JavaScript is the scripting language often used in the web development to validate the input of forms. It is a high level programming language that is characterised as dynamic and weakly typed.
- **SQL** – Structured Query Language is a domain specific language used for creating, updating, modifying and deleting the database. It is a standard language for relational database management systems.

The terminologies associated with the online portal for selling and buying of used books are:

- **Administrator** – The administrator controls the database, remove fake books, remove customers for improper behaviour etc.
- **User** – The user is the customer or the seller, a student who can buy and sell the books of a particular subject currently in possession of another student.
- **Portal** – It refers to the ubiquitous platform which provides the means for students to sell or buy the used books every semester at discounted rates.
- **Bookmark** – It refers to the wishlist which keeps the books liked by the user saved even when the session is closed after logging out.

1.2. Purpose

- The system provides a solution for purchasing of used books of different categories at cheaper and competitive prices.
- The secondary goal of the system is to provide recommendations and reviews for books for a particular subject.
- It is intended to reduce crowd, chaos and inevitable confusion among the college students for purchasing books every semester.

1.3. Motivation

- The motivation for the project was taken from the current scenario in colleges wherein students purchase costlier books every semester or rent them with very less returns. This project aims to solve this problem.
- The online portal will also help students to save their time and efforts by going to various shops and searching for books. It will also save the money of students by providing comparable prices for the same book from various sellers.
- Also taking up this project will help us to understand the subject theoretically and practically as well.
- This project will additionally help us in implementing the concepts learnt in web programming course. The technology used in this project is implemented in a similar way as in case of an industrial standard software project.

1.4. Problem Statement

- The software is designed to make the process of buying and selling of used books very simple. The database designed for online book reselling portal is primarily to achieve the functionalities of buying or selling of second hand books and get recommendations of books for a particular subject.

1.5. Objective

- It provides a hassle free solution for reselling and buying of used books for students and hence solves their problems.
- It aims to facilitate the students to provide recommendation and feedback on the books available on the platform.
- It aims to aid students to manage their book purchases and sales with regard to simplifying and speeding up the process of selection, ordering and purchasing books for buyers as well as managing a database of users and a database of products through a conveniently designed GUI which will utilise a user-friendly intuitive design approach.

1.6. Scope and Relevance

- The scope of this project lies majorly in school and college students but the conceptual scope can be extended further to the general public as well.
- The designed system keeps track of the details of users, books and transaction details.

Chapter 2: Requirement Specifications

2.1. Specifications

- The portal will provide a common facility to login as a user and perform the role of Buyer or Seller (or both).
- Browse through all the available books uploaded by the users.
- Smart and easy to find live search option.
- Register and modify the user profile.
- Add or delete the books from bookmark and save the session.
- Selecting Shipping method and mode of transaction.
- Provide review and recommendation for books.
- Update and delete the records based on user authentication.

2.1.1. Functional Requirements

The functional requirements define the function of a software or its components where a function is described as a specification of behavior between outputs and inputs. The application will support two different views with different functional roles mentioned as:

a) ADMIN View:

- Input: Admin Username and Password
- Process Definition: Login to the system
- Output: If the login is successful then the admin can:
 - viewBooks(): It will list all the books available on the portal with the Seller Name, Book Name, Author, Selling Price, Original Price, Edition. The admin can edit or delete any particular book.
 - viewUsers(): It will list all the users singed up on the portal along with the key information like Name, Email, City, Phone Number etc. The admin can edit or delete the information for any user.

- showReview(): It will list the reviews made by the user on any particular book. It will list the number of stars and the comments made by the user. The admin can edit the review or delete it.
- addCategory(): The admin can add, modify or delete the category available for books.

b) USER View:

- Input: Username and Password.
- Process Definition: Login to the system.
- Output: If the login is successful then the user can:
 - addBook(): The user can add book by filling all the necessary details. It will be visible on the homepage once successfully added to the database.
 - addBookmark(): The user can bookmark a particular book if it suits his interest. It is similar to the wishlist. It will be saved throughout the sessions. The user can also delete the bookmark.
 - addOrder(): The user can place the order by selecting a buy option and filling the transaction details.
 - register(): The new users can register by filing their personal information. The user's password will be saved in encrypted format for security purpose.
 - review(): The users can review a particular book and rate it as well on a scale of 1 to 5.

2.1.2. Non-Functional Requirements

The non-functional requirements are those which define the quality attribute of the software system. A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. The different quality standards used in this project are mentioned as:

a) Security Requirements

- The admin view is a protected view.
- Proper authentication via login form is required for viewing it.

- The password will not be shown in the URL.
 - A random number code will be sent to email for verifying the newly registered accounts.
- b) Performance Requirements
- Use of online fonts and styles have been used to a bare minimum to reduce the overhead time lag.
 - The livesearch option is very simple and provides suggestions based on the database entries.
- c) Stability
- The system is designed to be scalable to any e-commerce book store at any immediate demand with complete control on the book prices by the admin.
- d) SQL Injection
- Updating of details can be achieved only by pressing the update button. Just pasting the URL of submission won't update the profile.

2.1.3. User Interface Requirements

- Home Screen: The home screen will be a common interface and will display the books available and listed for sale. The avenues for login or sign up will be listed.
- Register Screen: The sign up form is made simple and easy to read by the users. It give a good appearance and is focussed.
- Book Details Screen: The details are placed adequately with the ratings provided in a graphical format for better appearance and easily comprehensible.

2.2. Hardware Requirements

The following are the minimum hardware requirements needed for the implementation of this project:

- CPU: Intel Core™, AMD Ryzen 5, 2.00 GHz or greater.
- RAM: 1GB DDR3
- Hard Drive: 5 GB+ free space

2.3. Software Requirements

The following are the software requirements for the project:

- OS: Windows 7/8/10 or Ubuntu Linux 18.0 LTS
- Database: MySQL 5.5 (XAMPP: MySQL with Apache Server)
- Database User: root@localhost
- Server Side Script: PHP
- Client Side Script: JS
- Languages: HTML, CSS, SQL
- Apache Version: 2.4.41
- Server Type: MariaDB
- PHP Version: 7.4.2

Chapter 3: Design

3.1. ER Diagram

3.1.1. E-R Diagram Representation

An entity–relationship diagram describes interrelated things of interest in a specific domain of knowledge [1]. A basic ER model is composed of entity types and specifies relationships that can exist between entities. The ER diagram for Online Books Buying and Selling Portal is shown in Figure 3.1.

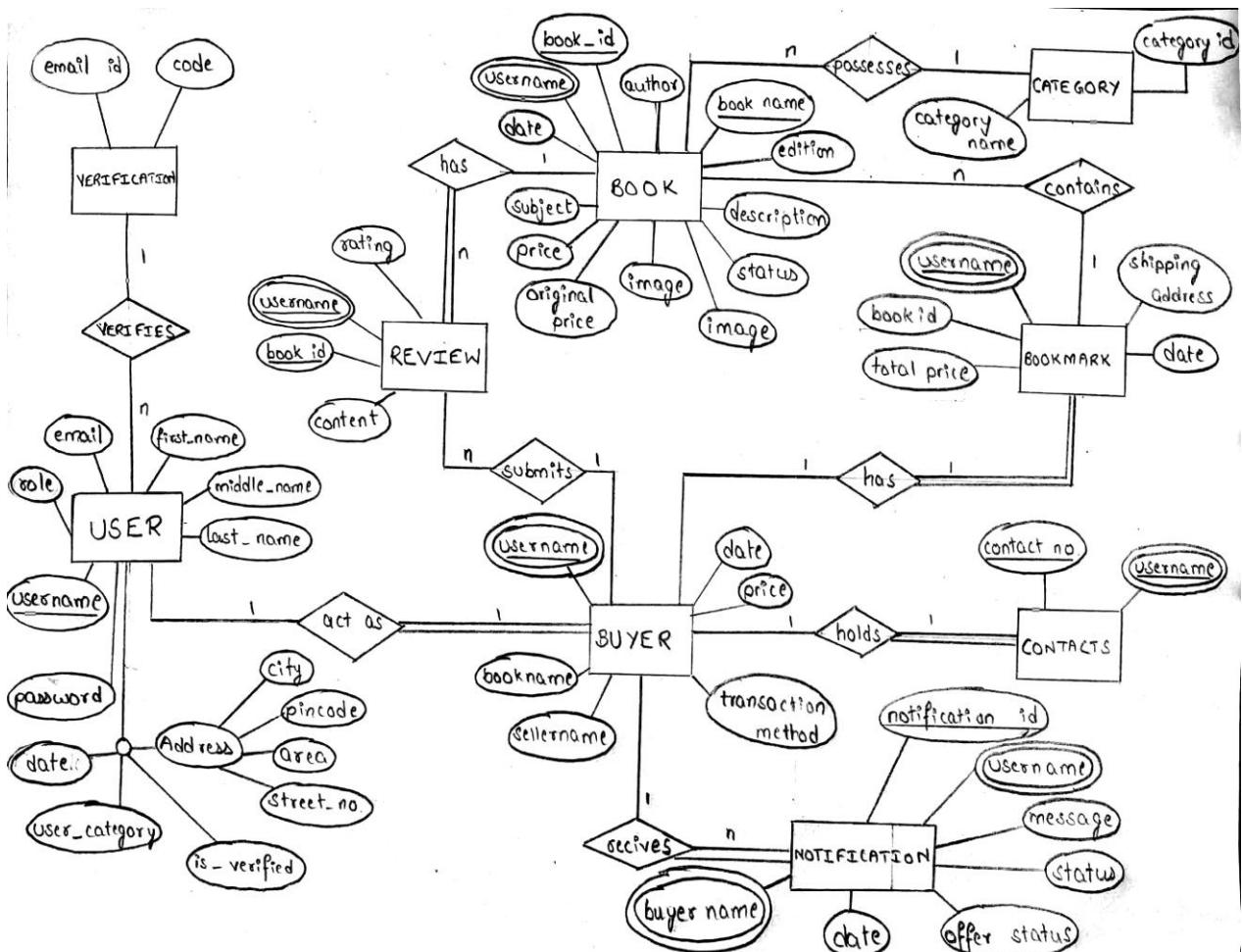


Figure 3.1: ER Representation for Online Platform

- The Primary Key attributes are shown by underline like username, book_id, user_id.

3.1.2. Schema Representation

The schema is the blueprint of how the database is constructed. The schema of online portal for buying and selling of books, normalised upto 3NF is shown in Figure 3.2.

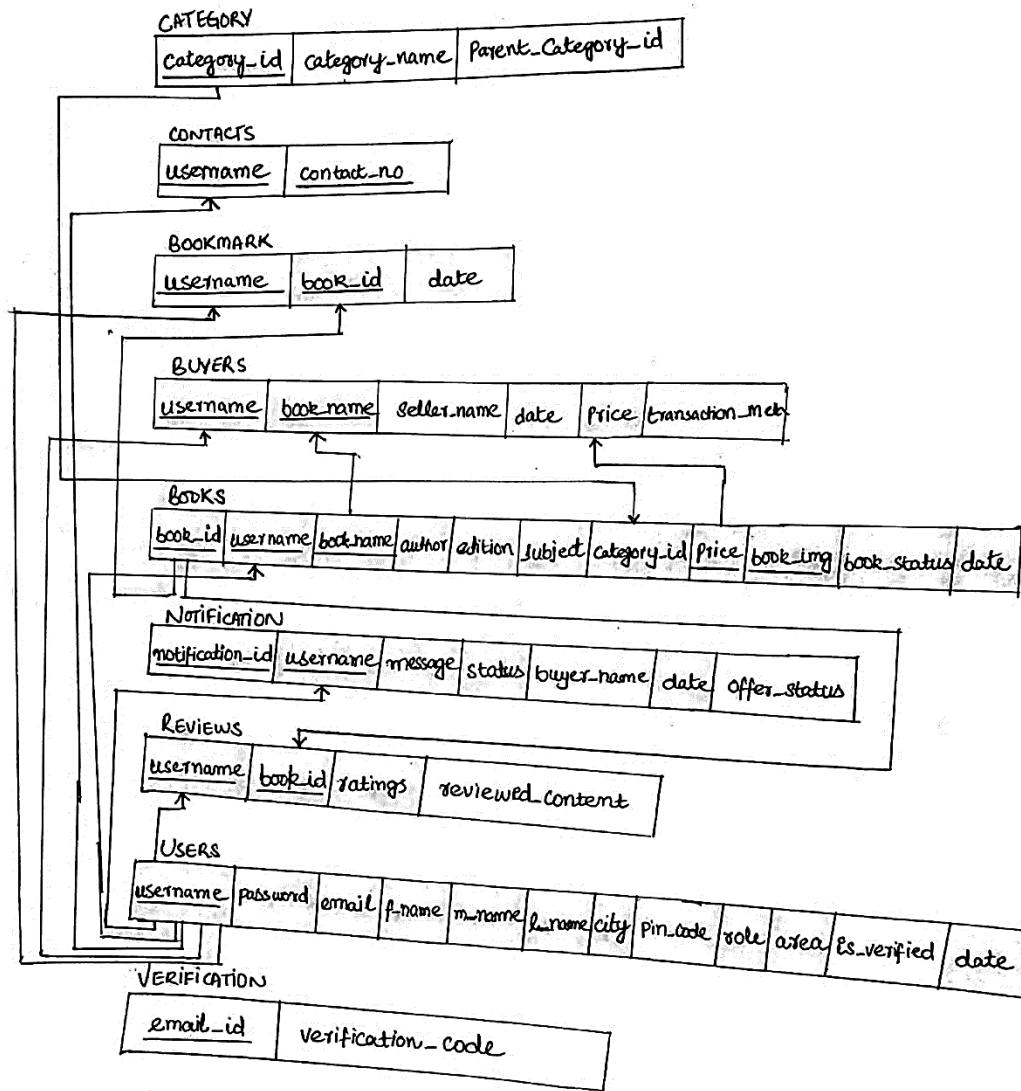


Figure 3.2: Schema Representation for Online Platform

- The username is the primary key for the table USERS and it serves as a foreign key to CONTACTS, BOOKMARKS, BUYERS, REVIEWS, NOTIFICATION and BOOKS relations.
- The book_id is the primary key in table BOOKS and foreign key for REVIEWS and BOOKMARKS relation.

3.2. UML Diagrams

3.2.1. Class Diagram

The class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects. The class diagram for the project is shown in Figure 3.3

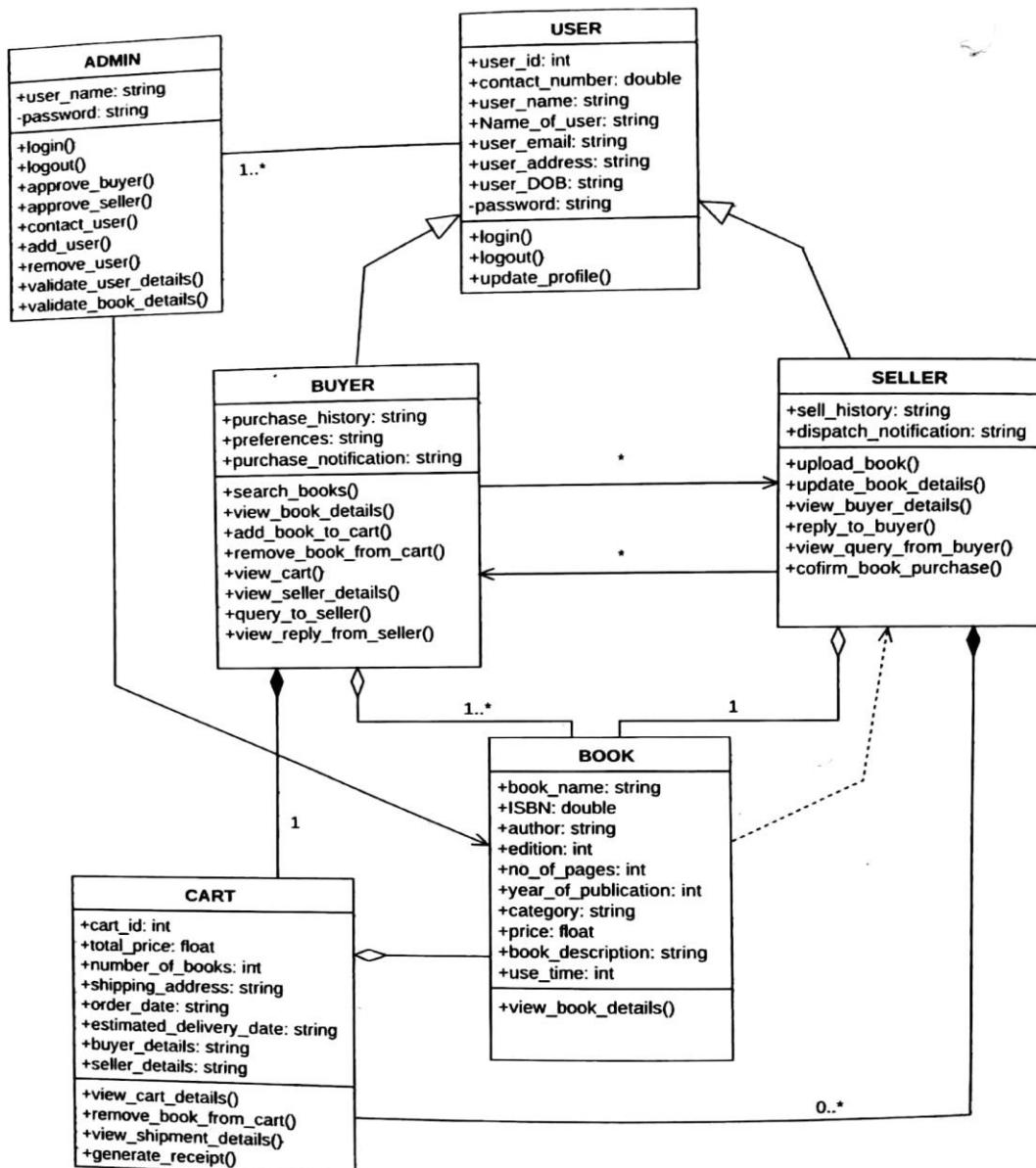


Figure 3.3: Class Diagram for Online Platform

- Classes BUYER and SELLER are the derived class from base class USER.
- Class CART has composition type of relationship with the BUYER and SELLER classes.

3.2.2. Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario [2].

The sequence diagrams for Online Book buying and selling platform are shown in Figure 3.4 and Figure 3.5.

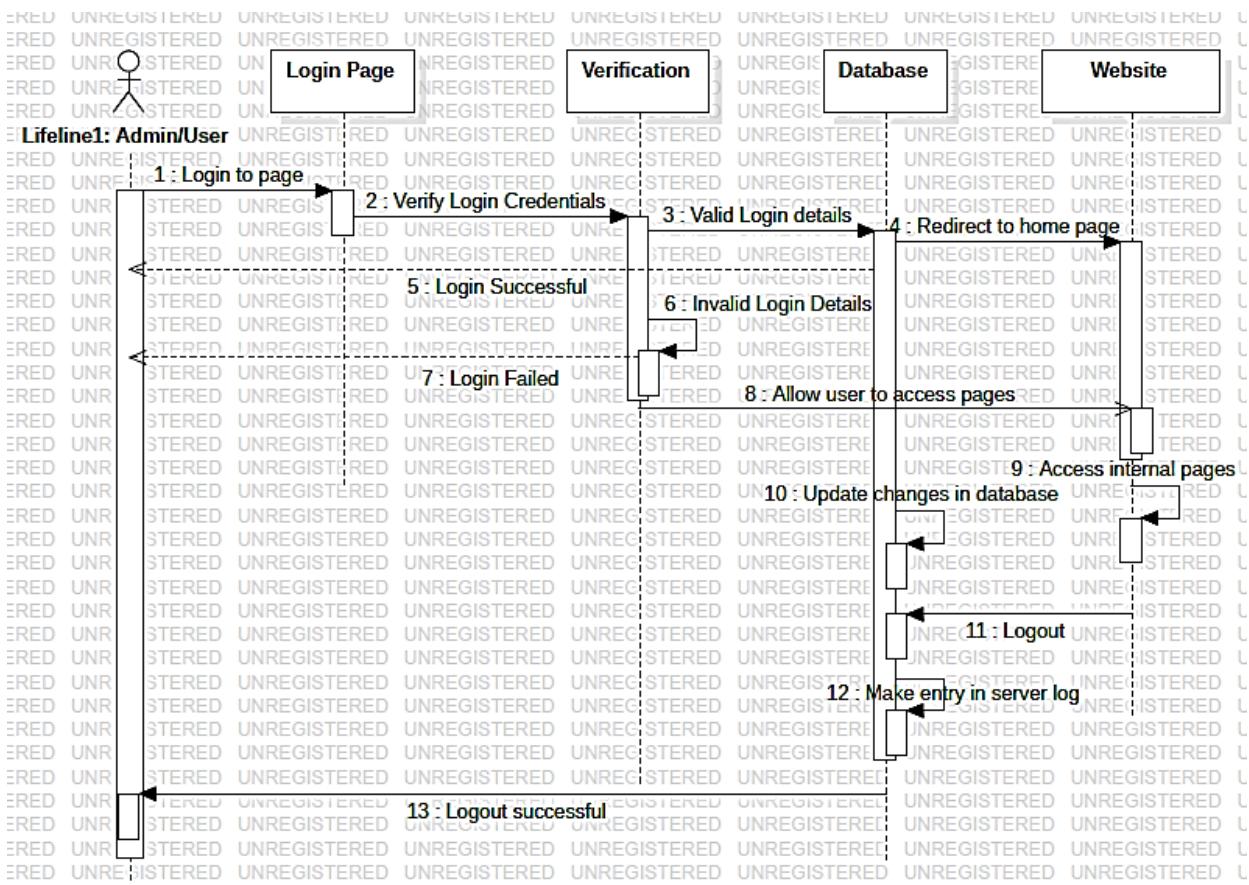


Figure 3.4: Sequence Diagram for Login Activity

- These diagrams portray the sequence of events from one state to another. It helps to track the user flow and mapping the overall working of a feature.

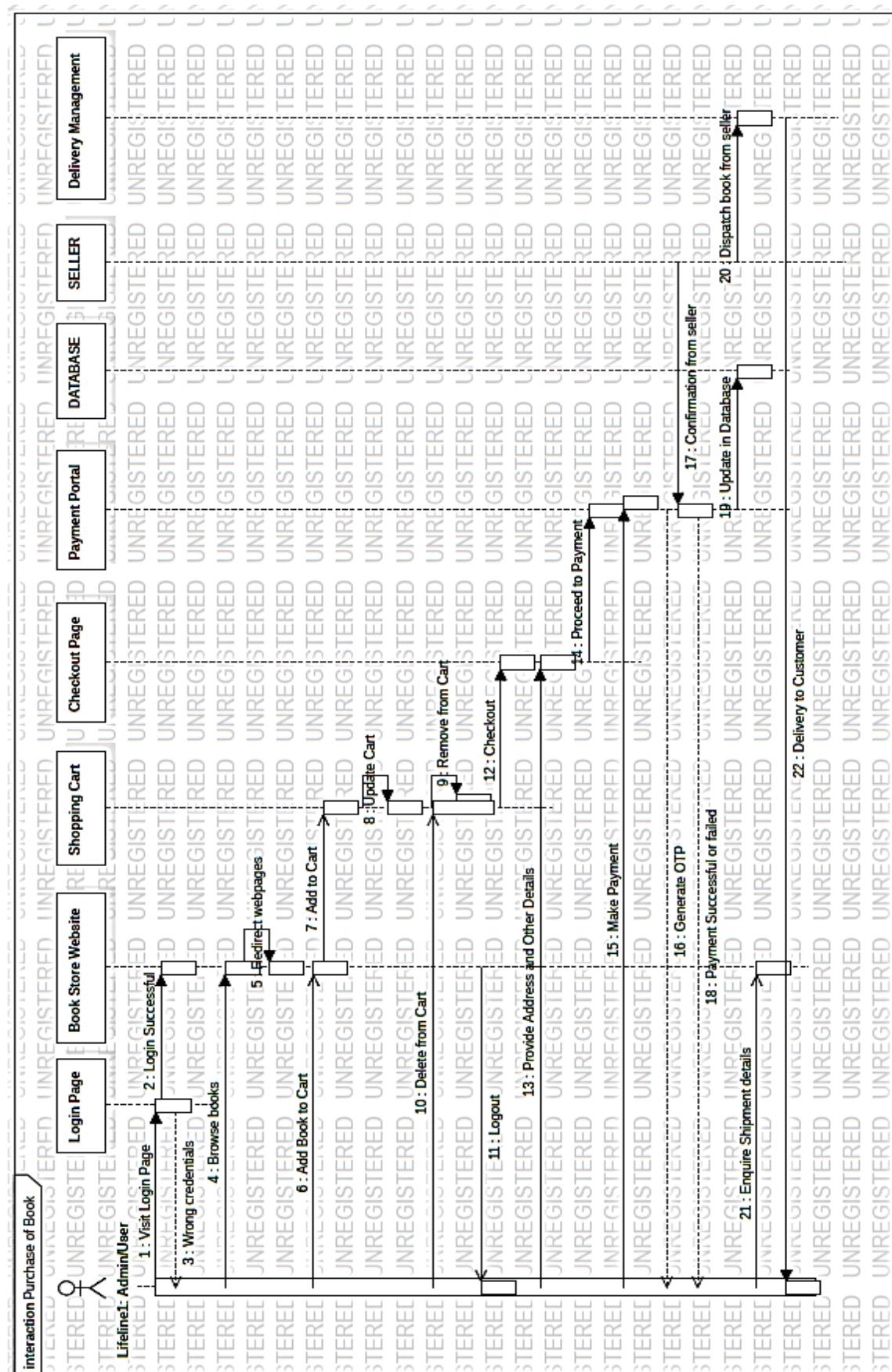


Figure 3.5: Sequence Diagram for Purchasing Activity

3.2.3. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The use case diagram for purchasing the book is shown in Figure 3.6

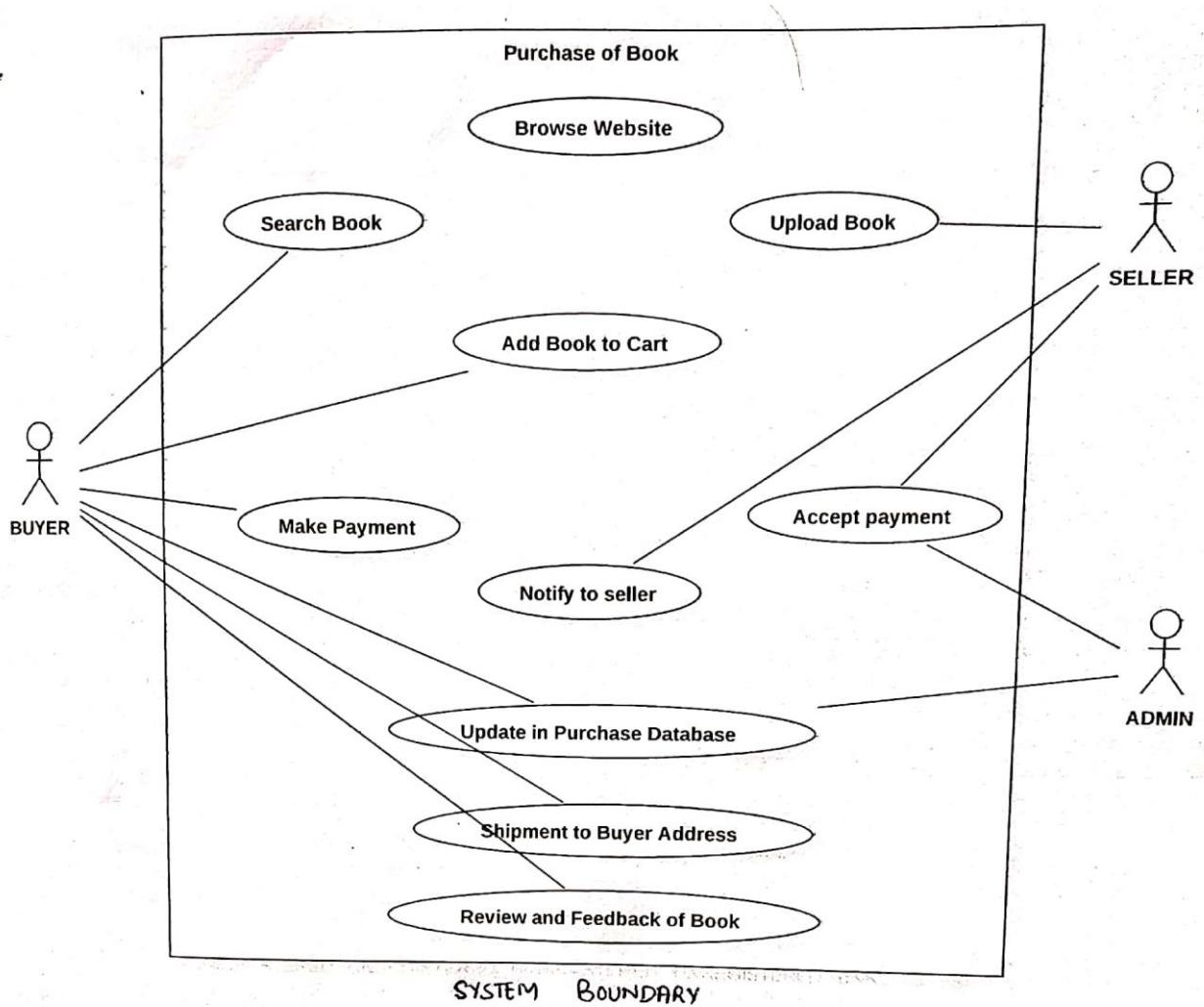


Figure 3.6: Use Case diagram for purchasing a book

- The portal will have three different users' view to support various functionalities, i.e. form the end of a buyer, seller or an admin.
- The admin cannot realize the features like adding a book to the cart and making purchases. These are restricted to only users.

The use case diagram for managing the shopping cart is shown in the Figure 3.7.

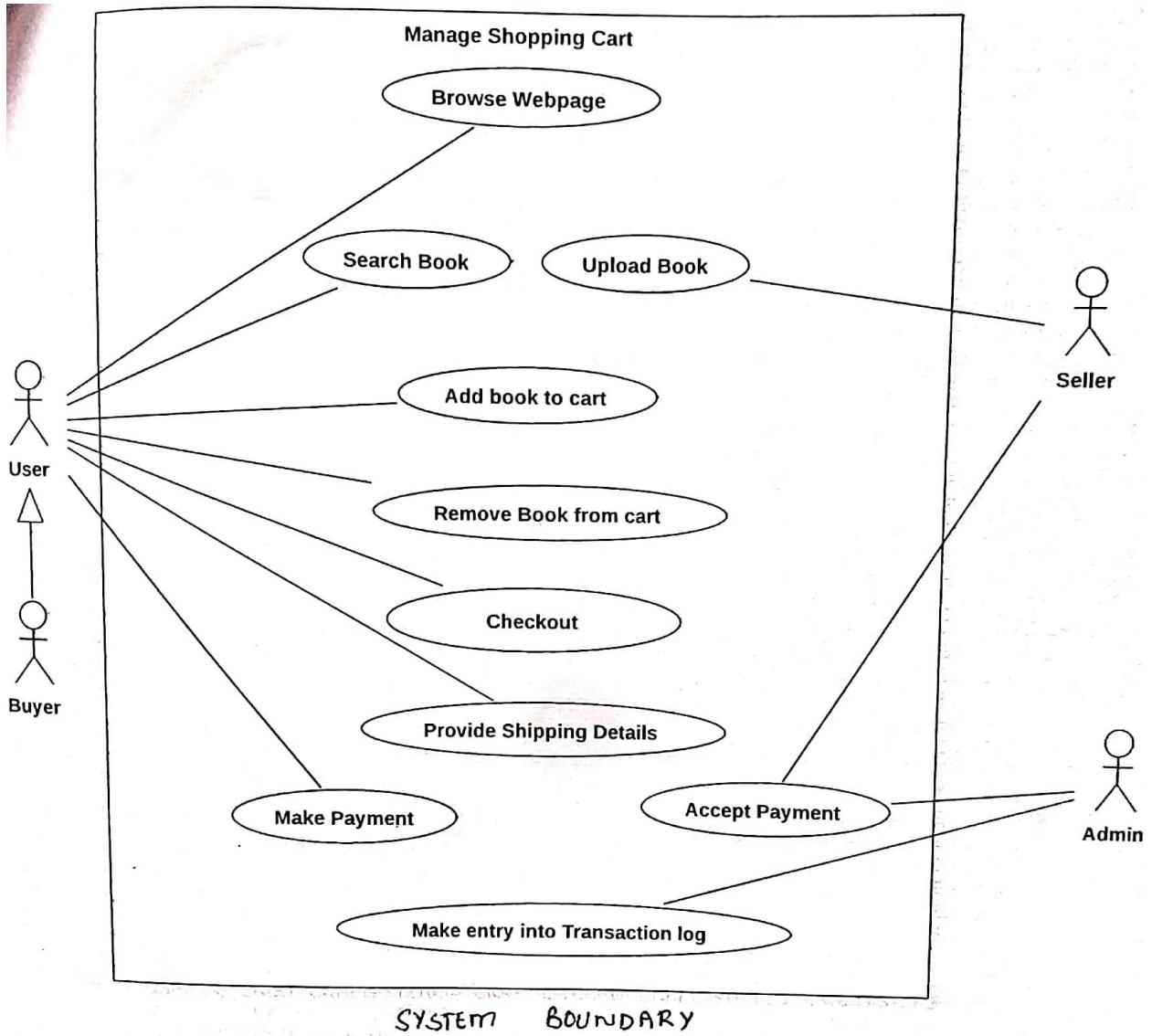


Figure 3.7: Use Case diagram for Shopping Cart

- The shopping cart is termed as ‘bookmark’ in this portal and the buyer is the main user for this feature.
- The user can save the books as a wishlist by keeping it bookmarked.

The use case diagram for maintaining of accounts and profile details is shown in Figure 3.8.

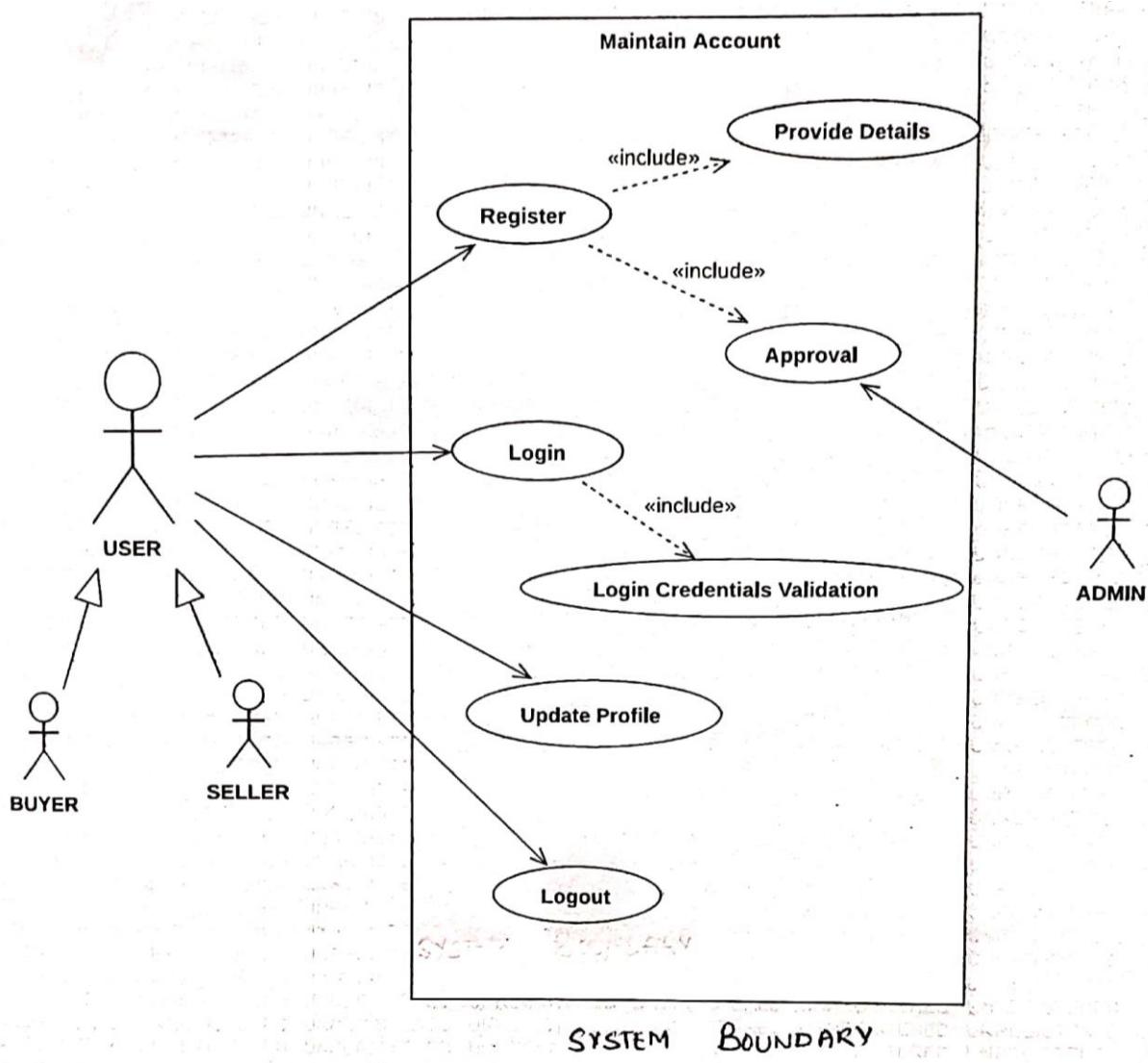


Figure 3.8: Use Case diagram for maintaining account

- The user is the common view for buyer and seller as they share a common account and hence have similar functionalities to look after.
- The other features supported are like login credentials verification and updating of user profile. The admin can add or delete any users.

3.2.4. Activity Diagram

Activity diagram describes the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The activity diagram for the admin is shown in Figure 3.9

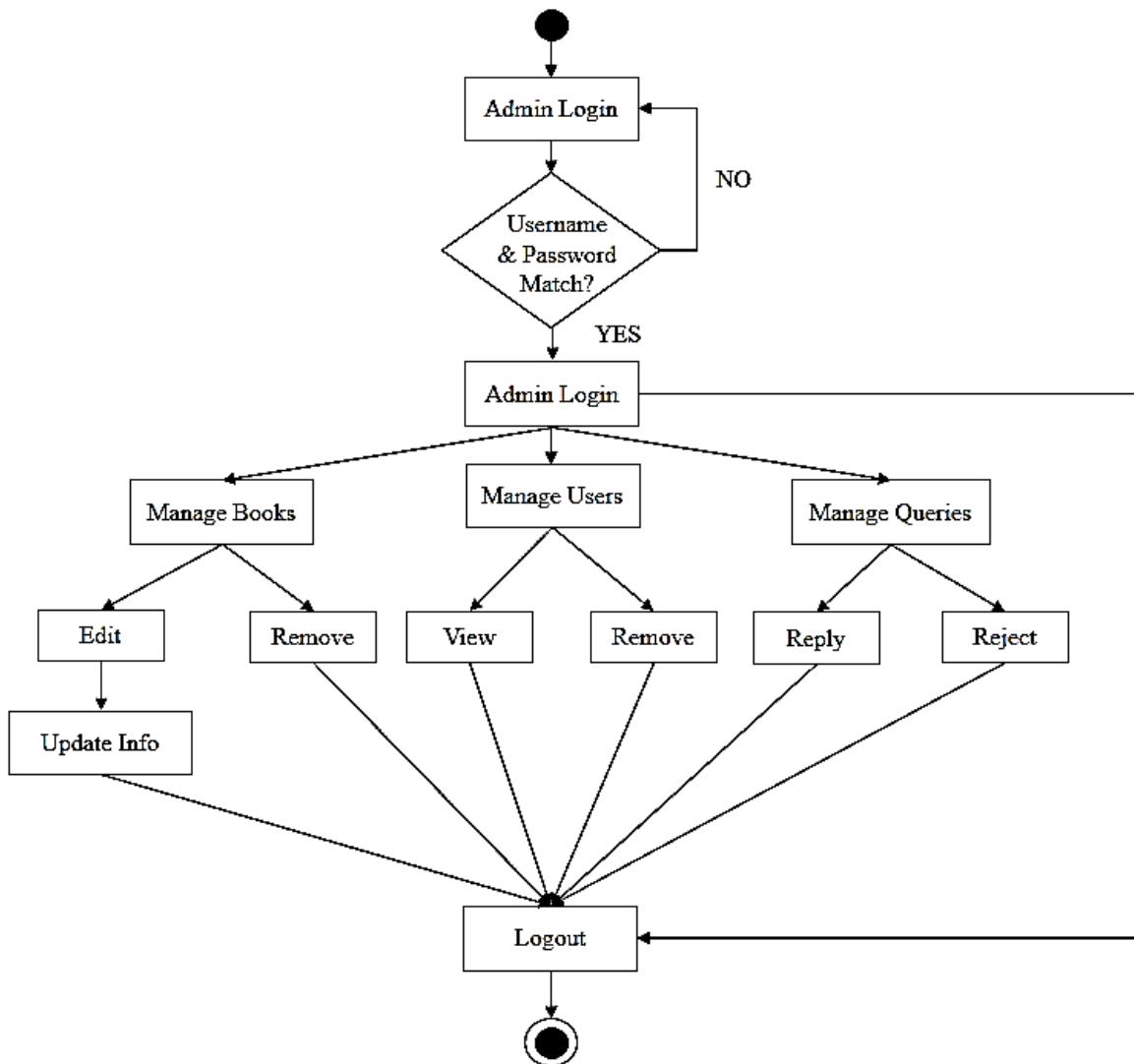


Figure 3.9: Activity Diagram for Admin View

The activity diagram for the user view is shown in the Figure 3.10

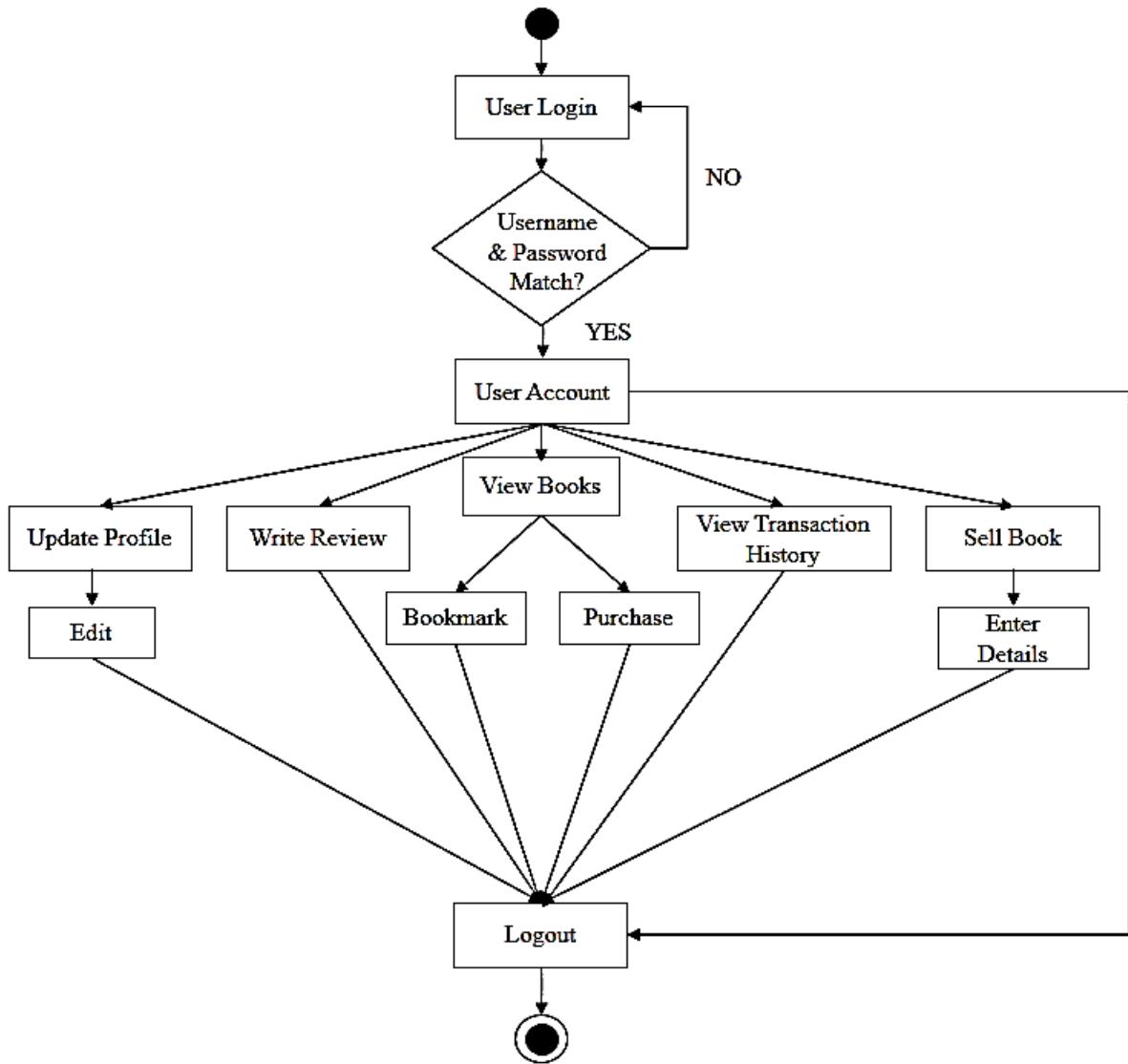


Figure 3.10: Activity diagram for User View

- The activity diagram for Admin and User view in given in Figure 3.9 and Figure 3.10.
- The activity starts with start of session and ends on logging off from the application.
- This diagram shows the dynamic view of the application created.

3.3. Front End Design

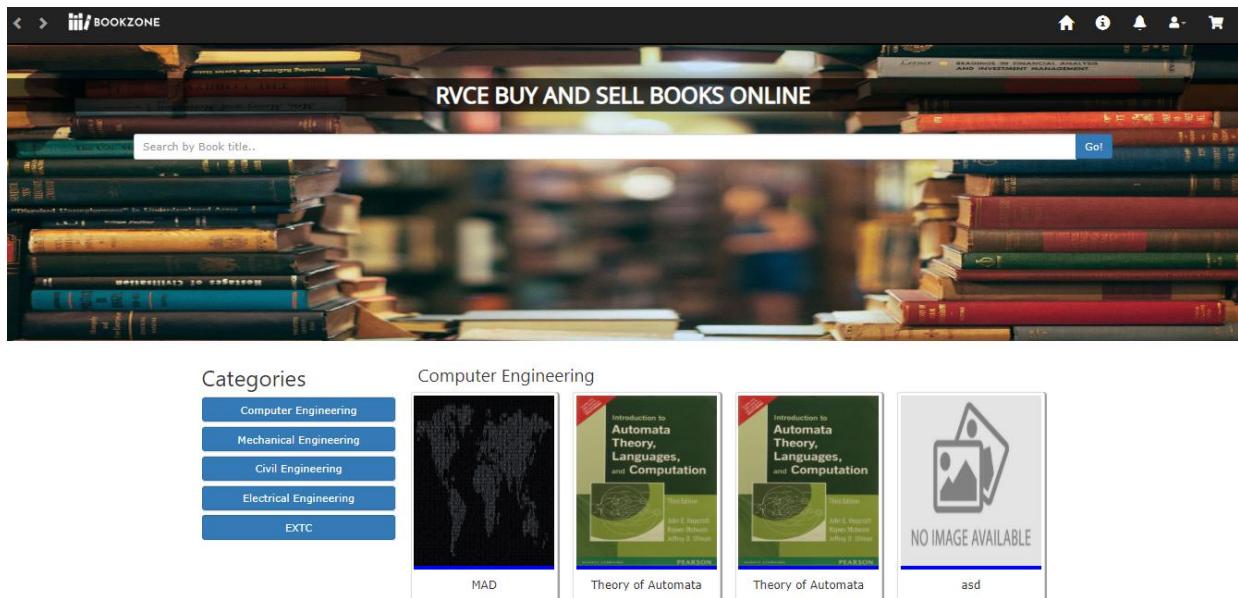


Figure 3.11: Home Page Front End Design

The screenshot shows a detailed view of a book listing. At the top, it says "Theory of Automata" by Rajeev Motwani. It shows the original price as ₹ 450 and the current price as ₹ 350 (22% off). There are buttons for "Buy Now" and "Bookmark". Below the book cover, which features the title and authors, are sections for "DESCRIPTION" and "DETAILED INFO". The "DESCRIPTION" section contains a paragraph about the book's content and updates. The "DETAILED INFO" section lists the subject as "Theory of Automata", category as "Computer Engineering", and provides details like the book name, author, and original price.

Figure 3.12: Book Details Front End Design

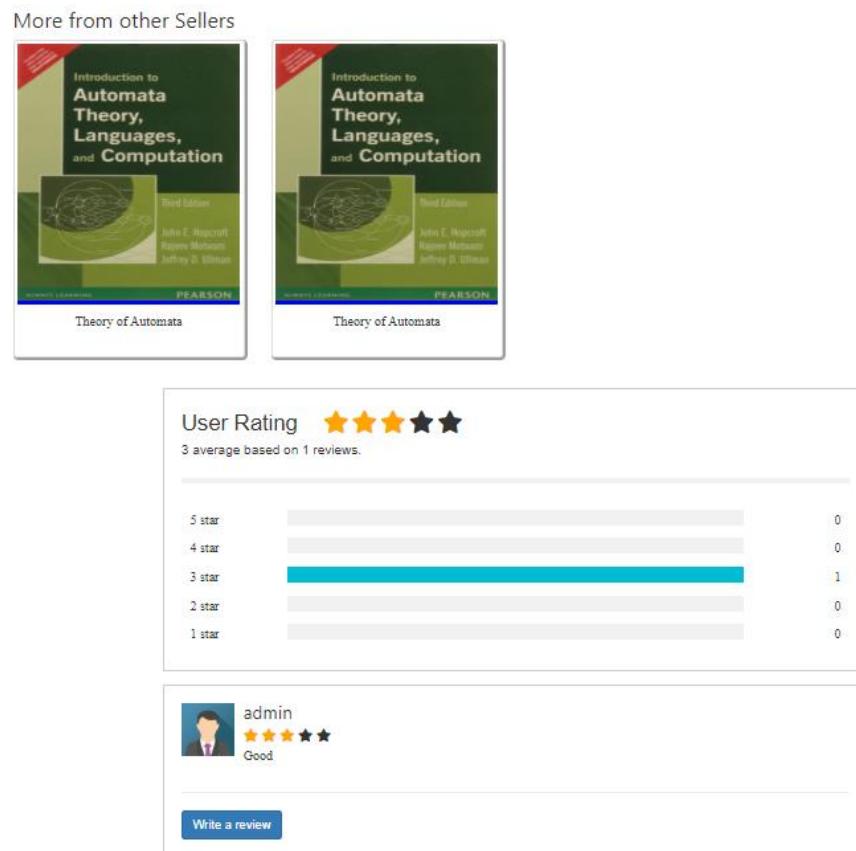


Figure 3.13: Review and User Ratings Front End

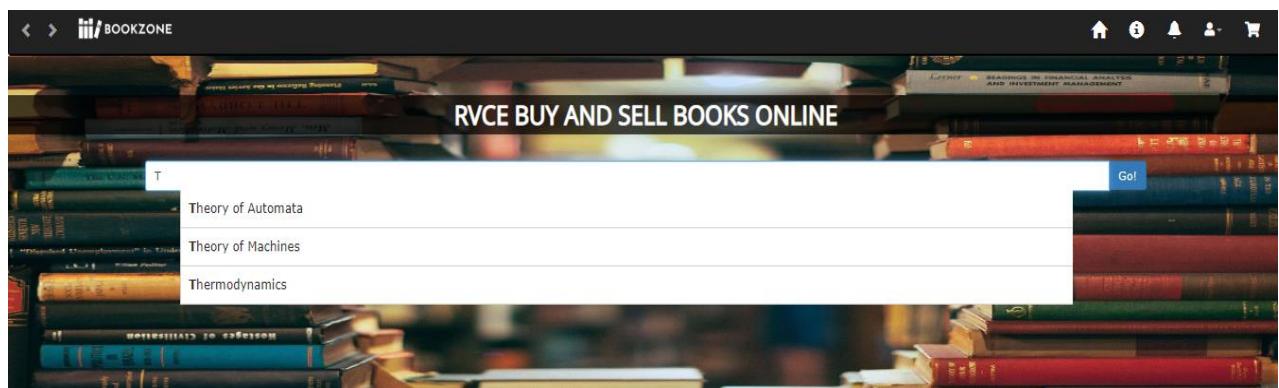


Figure 3.14: Smart search using keywords Front End

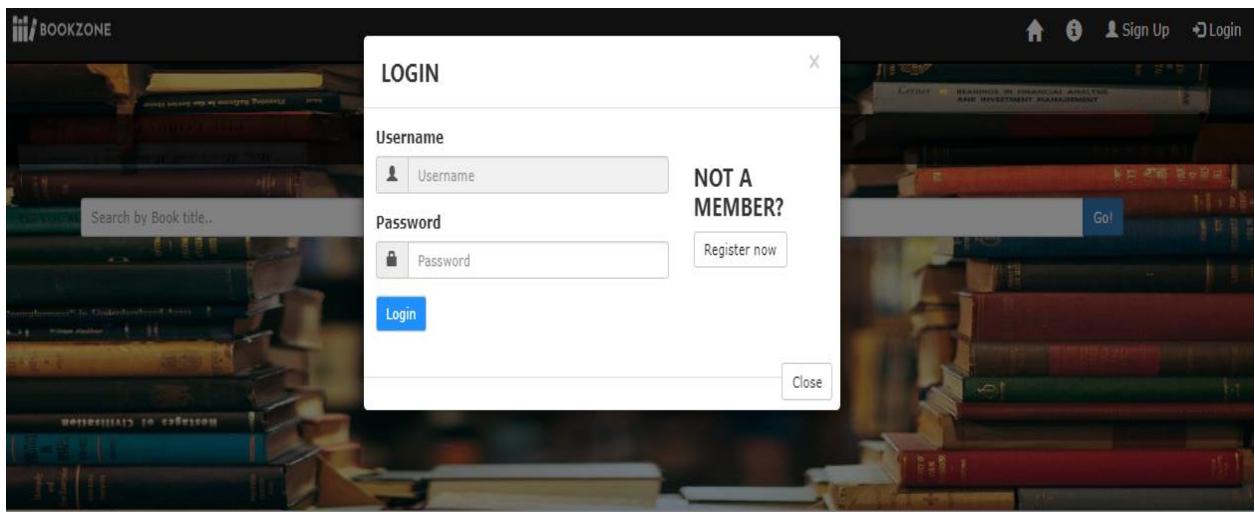


Figure 3.15: Login Page

The image displays the registration form for the BookZone portal. The title "Registration Form" is centered at the top. The form consists of several input fields: "Username" (with a person icon), "Password" (with a lock icon), "Confirm Password" (with a lock icon), and "Email" (with a person icon). Below these are three separate input fields for "First Name", "Middle Name", and "Last Name", each with a person icon. Further down are fields for "City" (with a location pin icon) and "Pincode" (with a location pin icon). On the right side of the form, there is a section titled "Your Preference" with a dropdown menu showing "Computer". At the bottom of the form is a blue "Verify and Register" button.

Figure 3.16: Sign up Form



Figure 3.17: Admin Dashboard

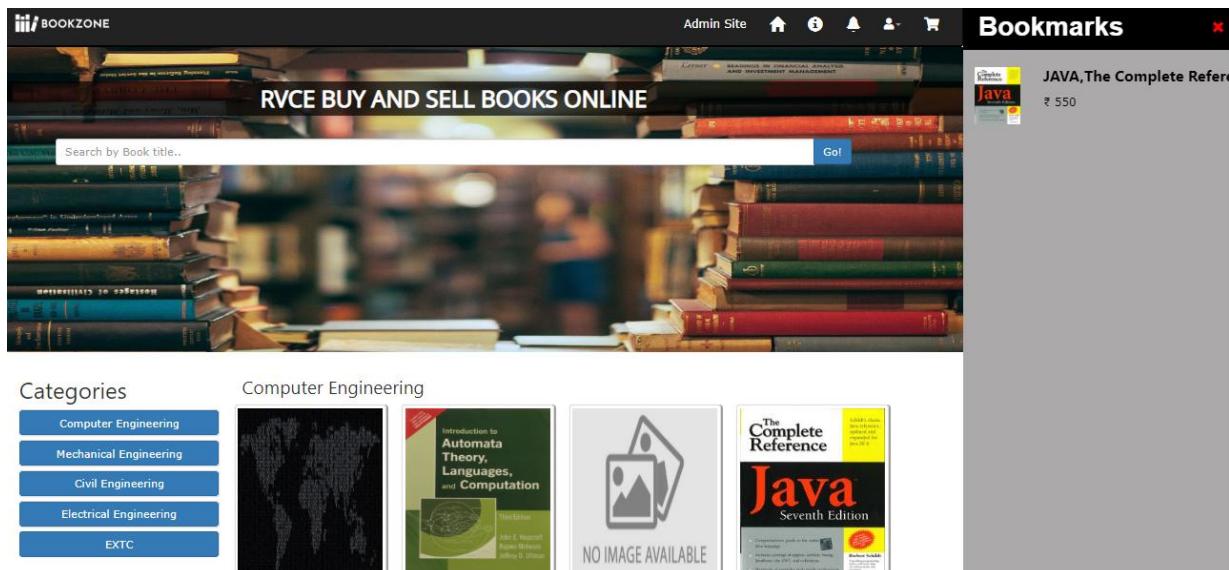


Figure 3.18: Added to Bookmark

Chapter 4: Implementation Details

4.1. Database Implementation

4.1.1. Table Creation

Following are the 9 tables created for this project:

- bookmark
- books
- buyers
- categories
- contacts
- notifications
- reviews
- users
- verification

- The code snippets associated with the table creation is:

```
CREATE TABLE `bookmark` (
  `username` varchar(255) NOT NULL,
  `book_id` int(11) NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `books` (
  `book_id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `book_name` varchar(255) NOT NULL,
  `author` varchar(255) NOT NULL,
  `edition` varchar(255) NOT NULL,
  `subject` varchar(255) NOT NULL,
  `category_id` int(11) NOT NULL DEFAULT '0',
  `book_price` int(11) NOT NULL DEFAULT '0',
  `book_original_price` int(11) NOT NULL DEFAULT '0',
  `book_description` varchar(1000) DEFAULT 'No Description available',
  `book_image` varchar(255) NOT NULL DEFAULT 'default.jpg',
  `book_status` varchar(255) NOT NULL DEFAULT 'available',
  `date` date NOT NULL DEFAULT '1920-05-20'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```

CREATE TABLE `buyers` (
  `username` varchar(255) NOT NULL,
  `book_name` varchar(255) NOT NULL,
  `seller_name` varchar(255) NOT NULL,
  `date` date NOT NULL,
  `price` float NOT NULL,
  `transaction_method` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `categories` (
  `category_id` int(11) NOT NULL,
  `category_name` varchar(255) NOT NULL,
  `parent_category_id` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `contacts` (
  `username` varchar(255) NOT NULL,
  `contact_no` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `notification` (
  `notification_id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `message` varchar(1000) NOT NULL,
  `status` varchar(255) NOT NULL DEFAULT 'Unseen',
  `buyer_name` varchar(255) DEFAULT NULL,
  `date` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `offer_status` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `users` (
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL DEFAULT 'noemail@noemail@noemail',
  `first_name` varchar(255) NOT NULL,
  `middle_name` varchar(255) DEFAULT NULL,
  `last_name` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `pincode` int(11) NOT NULL DEFAULT '-1',
  `user_category` varchar(255) NOT NULL DEFAULT 'FirstYear',
  `role` varchar(255) NOT NULL,
  `street_no` int(11) DEFAULT '-1',
  `area` varchar(255) DEFAULT NULL,
  `is_verified` varchar(25) DEFAULT NULL,
  `date` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `verification` (
  `email_id` varchar(255) NOT NULL,
  `code` int(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `reviews` (
  `username` varchar(255) NOT NULL,
  `book_id` int(11) NOT NULL,
  `ratings` int(11) NOT NULL DEFAULT '0',
  `review_content` varchar(255) NOT NULL DEFAULT 'No Review Content'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

- The code snippet for adding indexes to the dumped table is:

```

ALTER TABLE `bookmark`
ADD PRIMARY KEY (`username`, `book_id`),
ADD KEY `fk_book_id_bookmark` (`book_id`);

```

```

ALTER TABLE `books`
ADD PRIMARY KEY (`book_id`),
ADD KEY `fk_books_username` (`username`),
ADD KEY `book_price` (`book_price`),
ADD KEY `book_name` (`book_name`),
ADD KEY `book_image` (`book_image`);

```

```

ALTER TABLE `buyers`
ADD KEY `fk_username_users` (`username`),
ADD KEY `fk_book_name` (`book_name`);

```

```

ALTER TABLE `categories`
ADD PRIMARY KEY (`category_id`);

```

```

ALTER TABLE `contacts`
ADD PRIMARY KEY (`contact_no`),
ADD KEY `fk_username_contacts` (`username`);

```

```

ALTER TABLE `notification`
ADD PRIMARY KEY (`notification_id`),
ADD KEY `fk_username_users` (`username`);

```

```

ALTER TABLE `reviews`
ADD PRIMARY KEY (`username`, `book_id`),
ADD KEY `fk_book_id_reviews` (`book_id`);

```

```

ALTER TABLE `users`
ADD PRIMARY KEY (`username`);

ALTER TABLE `books`
MODIFY `book_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=34;

ALTER TABLE `categories`
MODIFY `category_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=17;

ALTER TABLE `notification`
MODIFY `notification_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=22;

```

- The code snippet for adding constraints to the dumped table is:

```

ALTER TABLE `bookmark`
ADD CONSTRAINT `fk_book_id_bookmark` FOREIGN KEY (`book_id`)
REFERENCES `books`(`book_id`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `fk_username_bookmark` FOREIGN KEY (`username`)
REFERENCES `users`(`username`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `books`
ADD CONSTRAINT `fk_books_username` FOREIGN KEY (`username`)
REFERENCES `users`(`username`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `contacts`
ADD CONSTRAINT `fk_username_contacts` FOREIGN KEY (`username`)
REFERENCES `users`(`username`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `reviews`
ADD CONSTRAINT `fk_book_id_reviews` FOREIGN KEY (`book_id`)
REFERENCES `books`(`book_id`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `fk_username_reviews` FOREIGN KEY (`username`)
REFERENCES `users`(`username`) ON DELETE CASCADE ON UPDATE CASCADE;

```

4.1.2. Table Population

- Following is the Code Snippet for Defining Local Procedures:

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `addUser` (IN
`param_username` VARCHAR(255), IN `param_password` VARCHAR(255), IN
`param_email` VARCHAR(255), IN `param(firstName` VARCHAR(255), IN
`param_middleName` VARCHAR(255), IN `param.lastName` VARCHAR(255), IN
`param_city` VARCHAR(255), IN `param_pincode` INT(11), IN
`param_category` VARCHAR(255), IN `param_role` VARCHAR(255), IN
`param_contact` VARCHAR(20), IN `param_isverified` VARCHAR(25), IN
`param_date` DATE) BEGIN
    INSERT INTO
users(username,password,email,first_name,middle_name,last_name,city,p
incode,user_category,role,is_verified,date)
VALUES(param_username,param_password,param_email,param(firstName,para
m_middleName,param_lastName,param_city,param_pincode,param_category,p
aram_role,param_isverified,param_date);
    INSERT INTO contacts(username,contact_no)
VALUES(param_username,param_contact);
END$$

```

- Following is the code snippet for dumping data into the respective tables:

```

INSERT INTO `books` (`book_id`, `username`, `book_name`, `author`,
`edition`, `subject`, `category_id`, `book_price`,
`book_original_price`, `book_description`, `book_image`,
`book_status`, `date`) VALUES
(1, 'murtaza', 'Kumbhojkar', 'Kumbhojkar', '6', 'Maths', 3, 250, 400,
'90% of his book contains questions which have already appeared on
previous years question papers. So it gives a wrong idea that
question papers of Mumbai University are set from his book. ',
'kumbhojkar.jpg', 'available', '2020-08-02'),
(3, 'ojas', 'Analysis of Algorithms', 'Sartaj Sahani', '5', 'AOA', 1,
275, 400, 'The first section of each of these chapters describes the
design method. The remaining sections apply the method to solve real
world applications. Allows students to compare the efficiency of the
solution resulting from the design method and some other methods',
'aoa.jpg', 'unavailable', '2018-08-15'),
(4, 'priyesh', 'Computer Architecture', 'William Stalling', '6',
'COA', 1, 499, 650, 'Four-time winner of the best Computer Science
and Engineering textbook of the year award from the Textbook and
Academic Authors Association, Computer Organization and Architecture:
Designing for Performance provides a thorough discussion of the
fundamentals of computer organization and architecture, covering not
just processor design, but memory, I/O, and parallel systems. Coverage
is supported by a wealth of concrete examples emphasizing modern
systems', 'coa.jpg', 'available', '2020-08-25');

```

```

INSERT INTO `categories` (`category_id`, `category_name`,
`parent_category_id`) VALUES
(1, 'Computer Engineering', 0),
(2, 'Mechanical Engineering', 0),
(3, 'Civil Engineering', 0),
(4, 'Electrical Engineering', 0),
(5, 'EXTC', 0);

INSERT INTO `contacts` (`username`, `contact_no`) VALUES
('adam1234', '992065413'),
('admin', '7738878906'),
('aidan1234', '8890022133'),
('arsitotle99', '8890133445'),
('baisa00', '7788713452'),
('basil55', '9992288721'),
('cameran12', '7764411221'),
('germaine21', '9876511111'),
('haley001', '7744532154'),
('hanan', '8655422442');

INSERT INTO `users` (`username`, `password`, `email`, `first_name`,
`middle_name`, `last_name`, `city`, `pincode`, `user_category`,
`role`, `street_no`, `area`, `is_verified`, `date`) VALUES
('adam1234', 'adam1234', 'adam@gmail.com', 'Adam', 'Lee', 'Wooten',
'Byculla', 2152, 'FirstYear', 'user', 13, 'diam nunc, ullamcorper eu,
euismod ac,', 'true', '2018-09-02'),
('admin', '$2y$10$wqeYW8EoWKStg2QKx/HZb.SIznv0tHSBmNW7onSsF6HCo080y0ia
e', 'ojask1205@gmail.com', 'admin', 'admin', 'admin', 'Thane',
400609, 'Computer', 'admin', -1, NULL, 'true', '2018-10-16'),
('aidan1234', 'aidan1234', 'aidan@gmail.com', 'Aidan', 'Wade',
'Wiley', 'Vikhroli', 34330, 'ThirdYear', 'user', 10, 'Vikhroli',
'true', '2018-09-20'), ('arsitotle99', 'arsitotle1234',
'arsitotle@gmail.com', 'Aristotle', 'Fulton', 'Duffy', 'Dadar', 9093,
'FourthYear', 'user', 20, 'Dadar', 'true', '2018-11-20'),
('baisa00', 'basia1234', 'basia@gmail.com', 'Basia', 'Wade', 'House',
'Kurla', 9448, 'FourthYear', 'user', 11, 'Kurla', 'true', '2018-11-
11'), ('basil55', 'basil1234', 'basil@gmail.com', 'Basil', 'Malik',
'Clemons', 'Charni Road', 595640, '', 'user', 15, 'Charni road',
'true', '2018-10-23'), ('cameran12', 'cameran1234',
'cameran@gmail.com', 'Cameran', 'Carter', 'Bartlett', 'Dahisar',
11271, '', 'user', 3, 'Dahisar', 'true', '2018-08-05'), ('germaine21',
'germaine1234', 'germaine@gmail.com', 'Germaine', 'Warren',
'Pollard', 'Churchgate', 968312, 'SecondYear', 'user', 4,
'Churchgate', 'true', '2018-11-18'), ('haley001', 'haley1234',
'haley@gmail.com', 'Haley', 'Leonard', 'Brock', 'Dombivali', 5111,
'FourthYear', 'user', 12, 'Dombivali', 'true', '2018-10-
21'), ('harsh', '$2y$10$jcX33pTISzFgScUIe/FF7uNVzVy8dmhzecWanEQz8PS7GOU
1UmsRW', 'harsh.patel14@somaiya.edu', 'harsh', 'j', 'patel',
'Nalasopara', 400609, 'Computer', 'user', -1, NULL, 'true', '2018-10-
16'), ('ivan76', 'ivan1234', 'ivan@gmail.com', 'Ivan', 'Richard',
'Gaines', 'Borivali', 6330, 'SecondYear', 'user', 2, 'Borivali',
'true', '2020-12-12');

```

4.1.3. Query Execution and Output

- SELECT * FROM ‘books’ WHERE book_price < 250

book_id	username	book_name	author	edition	subject	category_id	book_price	book_original_price
31	harsh	Mechanical Engineering Design	Keiss Nisbett	10	Mechanical Engineering Design	3	89	195
25	harsh	Fundamental of Fluid Mechanics	Rothmayour	7	Fundamental of Fluid Mechanics	2	130	250
19	murtaza	asd	asd	asd	asd	1	132	456
18	priyesh	Data Structures Using C	Reema Thareja	5	DS	4	150	300
8	tanay	Data Structures Using C	Reema Thareja	5	DS	2	230	300

Figure 4.1: Show Book with price less than Rs. 250

- SELECT * FROM ‘books’ WHERE username = ‘harsh’

book_id	username	book_name	author	edition	subject	category_id	book_price	book_original_price	book_description	book_image
7	harsh	Computer Architecture	William Stallling	6	COA	6	450	650	Four-time winner of the best Computer Science and ...	coa.jpg
13	harsh	JAVA,The Complete Reference	Herbert Schildt	5	OOPM	3	499	700	Fully updated for Java SE 8, Java: The Complete Re...	java.jpg
25	harsh	Fundamental of Fluid Mechanics	Rothmayour	7	Fundamental of Fluid Mechanics	2	130	250	Fundamentals of Fluid Mechanics offers comprehensi...	fluid.jpg
29	harsh	Theory of Automata	Rajeev Motwani	3	Theory of Automata	1	350	450	This classic book on formal languages, automata th...	Tcs.jpg
31	harsh	Mechanical Engineering Design	Keiss Nisbett	10	Mechanical Engineering Design	3	89	195	A textbook of machine design is useful for student...	Med.jpg

Figure 4.2: Show books from user named ‘Harsh’

- SELECT * FROM ‘contacts’ WHERE contact_no LIKE “9%”

username	contact_no
adam1234	992065413
basil55	9992288721
germaine21	9876511111
ivan76	9990077662
kylynn@301	9930111221
norman01	9002387672
quon@2212	9988788779
rebecca12	99922255566
sasha#22	9123188088
tanay	9987622111

Figure 4.3: List Contact Numbers which begin with number 9

- SELECT * FROM ‘users’ WHERE user_category = ‘Computer’

first_name	middle_name	last_name	city	pincode	user_category	role	street_no	area	is_verified
admin	admin	admin	Srinagar	560059	Computer	admin	-1	NULL	true
harsh	j	patel	Nalasopara	400609	Computer	user	-1	NULL	true
murtaza	t	patrawala	Kurla	400070	Computer	user	-1	NULL	true
ojas	suhas	kapre	Thane	123456	Computer	user	-1	NULL	true
priyesh		patel	Badlapur	410220	Computer	user	-1	NULL	true
tanay		raul	Mulund	400070	Computer	user	-1	NULL	true

Figure 4.4: List all users under category 'Computer'

- SELECT * FROM ‘categories’

category_id	category_name	parent_category_id
1	Computer Engineering	0
2	Mechanical Engineering	0
3	Civil Engineering	0
4	Electrical Engineering	0

Figure 4.5: List all available categories

4.2. Front End Implementation

4.2.1. Form Creation

Following forms have been implemented in the front end:

- Add Book (Figure 4.6)
- Review Form (Figure 4.7)
- Buy Now (Figure 4.8)
- Admin Edit Details (Figure 4.9)
- Register (Figure 4.10)
- Change Password (Figure 4.11)

```
<form method="POST" action="" onsubmit="return checkForm()" enctype="multipart/form-data">
    <div class="row">
        <div class="col-lg-6">
            <div class="form-group">
                <label for="book_category">Book Name</label>
                <div class="input-group">
                    <span class="input-group-addon"><i class="fa fa-book"></i></span>
                    <input name="book_name" type="text" class="form-control" placeholder="Enter book name" required="true">
                </div>
            </div>
            <div class="form-group">
                <label for="book_category">Selling Price</label>
                <div class="input-group">
                    <span class="input-group-addon"><i class="fa fa-inr"></i></span>
                    <input id="book_selling_price" onfocusout="checkPrice();" name="book_price" type="number" class="form-control" placeholder="Enter Selling price" required="true">
                </div>
            </div>
            <div class="form-group">
                <label for="book_category">Subject</label>
                <div class="input-group">
                    <span class="input-group-addon"><i class="fa fa-inr"></i></span>
                    <input name="book_subject" type="text" class="form-control" placeholder="Enter Subject" required="true">
                </div>
            </div>
            <div class="form-group">
                <label for="book_category">Edition</label>
                <input name="book_edition" type="number" class="form-control" placeholder="Enter book edition" required="true">
            </div>
        </div>
    </div>
```

```

<div class="col-lg-6">
    <div class="form-group">
        <label for="book_category">Author</label>
        <div class="input-group">
            <span class="input-group-addon"><i class="fa fa-user"></i></span>
            <input name="book_author" type="text" class="form-control" placeholder="Enter author name"
                   required="true">
        </div>
    </div>
    <div class="form-group">
        <label for="book_category">Original Price</label>
        <div class="input-group">
            <span class="input-group-addon"><i class="fa fa-inr"></i></span>
            <input id="book_original_price" onfocusout="checkPrice()" name="book_original_price"
                   type="number" class="form-control" placeholder="Enter Original price" required="true">
        </div>
    </div>
    <div class="form-group">
        <label for="book_category">Category</label>
        <select class="form-control" name="book_category">
            <option selected>Choose...</option>
            <?php
                $query = "SELECT * FROM categories WHERE parent_category_id = 0";
                $query_result_category = mysqli_query($connection,$query);
                if(!$query_result_category){
                    die("QUERY FAILED ".mysqli_error($connection));
                }
                while ($row_category = mysqli_fetch_assoc($query_result_category)) {
                    $category_id = $row_category['category_id'];
                    $category_name = $row_category['category_name'];
                    echo "<option value='".$category_id."'>".$category_name."</option>";
                }
            ?>
        </select>
    </div>
    <div class="form-group">
        <label for="title">Post Image</label>
        <input type="file" class="form-control" name="post_image">
    </div>
</div>
<div class="form-group">
    <label for="book_category">Book Description</label>
    <textarea id="editor" name="book_description" class="form-control" rows="10" placeholder="Enter book's description"></textarea>
</div>
<input name="submit" type="submit" class="btn btn-primary btn-center" value="Save Details">
</form>

```

Figure 4.6: Form for Adding Book

```

<form method="post" action="./templates/addReviews.php" id="reviewform">
    <input id="modal_book_id" type="text" value=<?php echo $_GET['book_id']; ?> name="book_id" style="display:none;">
    <input id="modal_book_username" type="text" value=<?php echo $_SESSION['username']; ?> name="username" style="display:none;">
    <div>
        <label><i class="fa fa-star"></i> Ratings</label>
        <select class="form-control" name="ratings">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
        </select>
    </div>
    <label>Comments</label>
    <textarea class="form-control" rows="5" name="review_content" form="reviewform" placeholder="Write a review!!! " style="width: 100%; resize: none;"></textarea>
    <input type="submit" class="btn btn-primary" name="review_submit">
</form>

```

Figure 4.7: Form for adding reviews

```

<form method="post" action="">
    <div class="form-group">
        <label for="sell1">Payment Method:</label>
        <select class="form-control" name="payment_method" id="sell1">
            <option>None</option>
            <option>PayTM</option>
            <option>Cash</option>
            <option>Net Banking</option>
            <option>Freecharge</option>
        </select>
    </div>

    <div class="form-group">
        <label for="sel2">Delivery Method:</label>
        <select class="form-control" name="delivery_method" id="sel2">
            <option>None</option>
            <option>Personal</option>
            <option>Courier</option>
        </select>
    </div>

    <input type="submit" id="order" class="btn btn-primary form-control" value="Order" name="order">
</form>

```

Figure 4.8: Form for transaction portal

```

<form method="post" action="">
    <div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
            <label for="email">Email</label>
            <div class="input-group">
                <span class="input-group-addon"><i class="fa fa-envelope"></i></span>
                <input type="Email" class="form-control" id="email" value="php echo $username?&gt;" placeholder="Email ID" disabled&gt;
            &lt;/div&gt;
            &lt;div class="row"&gt;
                &lt;div class="col-sm-4"&gt;
                    &lt;label for="fname"&gt;First Name&lt;/label&gt;
                    &lt;div class="input-group"&gt;
                        &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-user"&gt;&lt;/i&gt;&lt;/span&gt;
                        &lt;input type="text" class="form-control" id="fname" value="<?php echo $firstName?&gt;" placeholder="First name" disabled&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
                &lt;div class="col-sm-4"&gt;
                    &lt;label for="mname"&gt;Middle Name&lt;/label&gt;
                    &lt;div class="input-group"&gt;
                        &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-user"&gt;&lt;/i&gt;&lt;/span&gt;
                        &lt;input type="text" class="form-control" id="mname" value="<?php echo $middleName?&gt;" placeholder="Middle name" disabled&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
                &lt;div class="col-sm-4"&gt;
                    &lt;label for="lname"&gt;Last Name&lt;/label&gt;
                    &lt;div class="input-group"&gt;
                        &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-user"&gt;&lt;/i&gt;&lt;/span&gt;
                        &lt;input type="text" class="form-control" id="lname" value="<?php echo $lastName?&gt;" placeholder="Last name" disabled&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;

        &lt;div class="row"&gt;
            &lt;div class="col-sm-6"&gt;
                &lt;label for="flatNo"&gt;Flat No.&lt;/label&gt;
                &lt;div class="input-group"&gt;
                    &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-address-book"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;input name="street_no" type="number" class="form-control" id="flatNo" value="<?php echo $flatNo?&gt;" placeholder="Flat no. , Street no."&gt;
                &lt;/div&gt;
                &lt;label for="area"&gt;Area&lt;/label&gt;
                &lt;div class="input-group"&gt;
                    &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-address-book"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;input name="area" type="text" class="form-control" id="area" value="<?php echo $area?&gt;" placeholder="Area"&gt;
                &lt;/div&gt;
                &lt;label for="contact"&gt;Contact&lt;/label&gt;
                &lt;div class="input-group"&gt;
                    &lt;span class="input-group-addon"&gt;&lt;i class="glyphicon glyphicon-phone"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;input name="contact" type="number" class="form-control" id="contact" value="<?php echo $contactNo?&gt;" placeholder="Contact Number"&gt;
                &lt;/div&gt;&lt;br&gt;
            &lt;/div&gt;

            &lt;div class="col-sm-6"&gt;
                &lt;label for="city"&gt;City&lt;/label&gt;
                &lt;div class="input-group"&gt;
                    &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-address-book"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;input name="city" type="text" class="form-control" id="city" value="<?php echo $city?&gt;" placeholder="City"&gt;
                &lt;/div&gt;
                &lt;label for="pincode"&gt;Pincode&lt;/label&gt;
                &lt;div class="input-group"&gt;
                    &lt;span class="input-group-addon"&gt;&lt;i class="fa fa-address-book"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;input name="pincode" type="text" class="form-control" id="pincode" value="<?php echo $pincode?&gt;" placeholder="Pincode"&gt;
                &lt;/div&gt;
                &lt;div class="form-group"&gt;
                    &lt;label for="preference"&gt;Category&lt;/label&gt;
                    &lt;select name="category" class="form-control" id="preference"&gt;
                        &lt;option &amp;gt;?php if($category=='FirstYear'){echo 'selected';}&amp;gt; ?&gt; value="FirstYear"&gt;FirstYear
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/form&gt;
</pre

```

```

        <option value="SecondYear">SecondYear</option>
        <option <?php if($category=='ThirdYear'){echo 'selected';} ?> value="ThirdYear">ThirdYear</option>
        <option <?php if($category=='LastYear'){echo 'selected';} ?> value="LastYear">LastYear</option>
    </select>
</div>
</div>
<div class="row">
    <div class="col-sm-6 form-group">
        <input name="submit" type="submit" class="form-control btn btn-primary" value="Save Changes">
    </div>
    <div class="col-sm-6 form-group">
        <a data-toggle="modal" href="#" data-target="#changePwdModal" class="btn btn-primary form-control">
            Change Password
        </a>
    </div>
</div>
<div class="col-sm-3"></div>
</div>
</form>

```

Figure 4.9: Form for Editing User Details

```

<form method="post" onsubmit="return checkForm()" action=".//register.php">
<div class="row">
    <div class="col-sm-3"></div>
    <div class="col-sm-6">
        <div class="form-group">
            <label for="username">Username</label>
            <div class="input-group">
                <span class="input-group-addon"><i class="fa fa-envelope"></i></span>
                <input onkeyup="checkUsername(this.value)" name="username" type="text" class="form-control" id="username" placeholder="Username" required="true">
            </div>
            <p id="username_error" style="color:red;display:none;"><b>Sorry!! This username is already taken</b></p>
        </div>
        <div class="form-group">
            <label for="passwd">Password</label>
            <div class="input-group">
                <span class="input-group-addon"><i class="fa fa-lock"></i></span>
                <input name="password" type="password" class="form-control" id="passwd" placeholder="Password" required="true">
            </div>
        </div>
        <div class="form-group">
            <label for="confirmpwd">Confirm Password</label>
            <div class="input-group">
                <span class="input-group-addon"><i class="fa fa-lock"></i></span>
                <input onfocusout="checkPassword()" name="confirmpassword" type="password" class="form-control" id="confirmpwd" placeholder="Re-type password" required="true">
            </div>
            <p id="error_msg_confirmpwd" style="color:#F00; padding-top: 5px; display:none">ERROR!!! Password doesn't match</p>
        </div>
        <div class="form-group">
            <label for="email">Email</label>
            <div class="input-group">
                <span class="input-group-addon"><i class="fa fa-lock"></i></span>
                <input onkeyup="checkEmail(this.value)" name="email" type="Email" class="form-control" id="email" placeholder="Enter Email" required="true">
            </div>
            <p id="email_error" style="color:red;display:none;"><b>This email is already used, Please try some other email</b></p>
        </div>
    <div class="row">
        <div class="col-sm-4">
            <div class="form-group">
                <label for="fname">First Name</label>
                <div class="input-group">
                    <span class="input-group-addon"><i class="fa fa-user"></i></span>

```

```

<input onkeyup="checkEmail(this.value)" name="email" type="Email" class="form-control" id="email" placeholder="Enter Email
" required="true">
</div>
<p id="email_error" style="color:red;display:none;"><b>This email is already used, Please try some other email</b></p>
</div>
<div class="row">
<div class="col-sm-4">
<div class='form-group'>
<label for="fname">First Name</label>
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-user"></i></span>
<input name="firstname" onfocusout="checkName(this.value)" type="text" class="form-control" id="fname" placeholder
="First name" required="true">
</div>
</div>
</div>
<div class="col-sm-4">
<div class='form-group'>
<label for="mname">Middle Name</label>
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-user"></i></span>
<input name="middlename" onfocusout="checkName(this.value)" type="text" class="form-control" id="mname"
placeholder="Middle name">
</div>
</div>
</div>
<div class="col-sm-4">
<div class='form-group'>
<label for="lname">Last Name</label>
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-user"></i></span>
<input name="lastname" onfocusout="checkName(this.value)" type="text" class="form-control" id="lname" placeholder
="Last name" required="true">
</div>
</div>
</div>
<p id="name_error" style="color:red;display:none;"><b>Name cannot contain numbers!!!</b></p>
</div>
<div class="row">
<div class="col-sm-6">
<div class='form-group'>
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-address-book"></i></span>
<input name="city" type="text" class="form-control" id="city" placeholder="City" required="true">
</div>
<div class='form-group'>
<label for="contact">Contact</label>
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-phone"></i></span>
<input name="contact" onfocusout="checkContact(this.value);;" type="number" class="form-control" id="contact"
placeholder="Contact Number" valid><br>
</div>
</div>
<p id="contact_error" style="color:red;"></p>
</div>
<div class="col-sm-6">
<div class="form-group">
<label for="pincode">Pincode</label>
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-address-book"></i></span>
<input name="pincode" onfocusout="checkPincode(this.value)" type="number" class="form-control" id="pincode"
placeholder="Pincode" required="true" valid>
</div>
</div>
<p id="pincode_error" style="color:red;display:none;"><b>Enter 6 digit pincode!!!</b></p>
<div class="form-group">
<label for="preference">Your Preference</label>
<select name="user_category" class="form-control" id="preference">
<option value="Computer">Computer</option>
<option value="Mechanical">Mechanical</option>
<option value="IT">IT</option>
<option value="Electronics">Electronics</option>
<option value="EXTC">EXTC</option>
</select>
</div>
</div>
<div class='form-group'>
<input name="submit" type="submit" id="final_submit" class="btn btn-primary" value="Verify and Register">
</div>
</div>

```

Figure 4.10: Form for registering new user

```

<form method="post" action="./admin_changePasswd.php">
    <input type="text" name="username" style="display: none" value="<?php echo $username?>">
    <div class="form-group">
        <label for="oldPasswd">Old Password</label>
        <div class="input-group">
            <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
            <input id="oldPasswd" type="password" class="control" name="oldPasswd" placeholder="Old Password">
        </div>
    </div>

    <div class="form-group">
        <label for="newPasswd">New Password</label>
        <div class="input-group">
            <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
            <input id="newPasswd" type="password" class="form-control" name="newPasswd" placeholder="New Password">
        </div>
    </div>

    <div class="form-group">
        <label for="password">Confirm Password</label>
        <div class="input-group">
            <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
            <input id="comfirmPasswd" type="password" class="form-control" name="comfirmPasswd" placeholder="Confirm Password">
        </div>
    </div>
</div>
</form>

```

Figure 4.11: Form for changing password

4.2.2. Connectivity to Database

To access data in MySQL database, database connectivity to the server is required. mysqli is used to connect to the MySQL database and select the database. The PHP code to connect to database is shown in Figure 4.12.

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$database_name = "books-portal";

// Create connection
$connection = mysqli_connect($servername, $username, $password,$database_name);

// Check connection
if (!$connection) {
    echo "<script type='text/javascript'> alert(Retry, Connection Error!);";
    window.location = 'index.php';
    </script>";
    die("Connection failed: " . mysqli_connect_error());
}

?>

```

Figure 4.12: Connecting to database

4.2.3. Report Generation

The report is generated after the completion of user activity. Example:

- The user gets a unique verification code for successful account creation.
- The admin is also notified about the message from any seller on enquiry.
- The buyer is reported back on successful completion of transaction after purchasing the book.
- Once the review is submitted, the buyer can query the table and derive all the reviews about the book.
- The buyer can get a report of his purchase history.
- Only the admin has the privilege to generate the report for the list of transactions.
- The admin can generate the report for list of all available book and all active users on the portal.

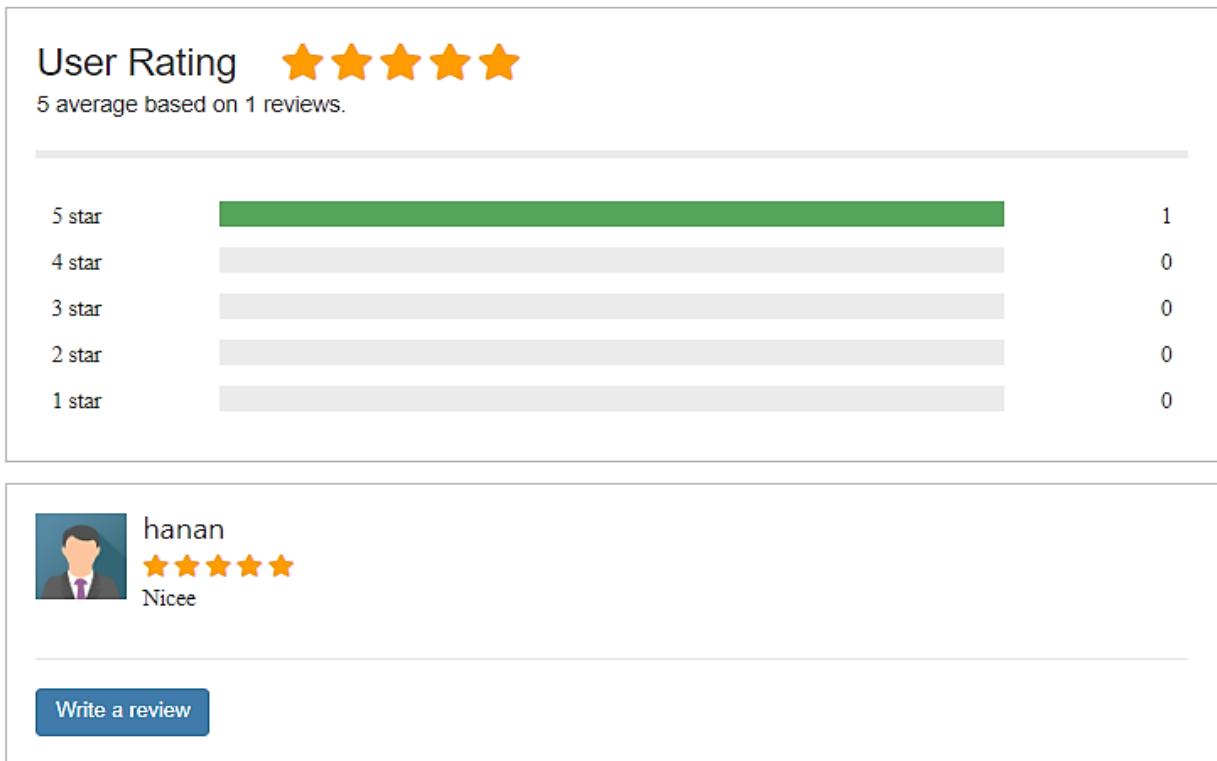


Figure 4.13: Sample report generation to view all reviews

<u>ALL TRANSACTIONS</u>					
Show 10 entries		Search:			
Buyer name	Seller name	Book name	Price	Date	Method
admin	harsh	Theory of Automata	350	2020-04-17	PayTM
hanan	ojas	Theory of Automata	399	2020-04-20	Cash

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 4.14: Report generation for Transactions

4.2.4. Security Features

- Once a person logs out and when they try to go back they will be logged out and login page is displayed immediately.
- If a user tries to access any web page through URL without logging in, it is not allowed and it will ask the user to enter login credentials.
- When the user enters their password it encrypts the password using md5 and store it in database.
- Any application user should get the admin privilege only after authentication.
- The list of transactions and details about personal information of users is available to only admin.

Chapter 5: Testing and Results

5.1. Database Testing

Database Testing is a type of software testing that checks the schema, tables, triggers etc. of the database under test. It involves creating complex queries for performing the load or stress test on the database and check its responsiveness. This part of testing includes checking and validating data present in the database while updating and inserting values into it to satisfy functional requirements of the system.

It also checks the data integrity and consistency. The following verifications are carried out during database testing: Checking the data Mapping, ACID (Atomicity, Consistency, Isolation, Durability) properties validation, Data Integrity, Business rule conformance [3].

5.1.1. Test Cases

The test cases identified and the results of each test case is shown in Table 5.1.

Table 5.1: Test Cases for Database Testing

TESTCASE ID	TESTCASES	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TSD01	Add any book to Bookmark	Added Successfully	Added and listed in bookmarks	Pass
TSD02	Delete a Bookmark from database	Deleted Successfully	Book not visible in bookmarks	Pass
TSD03	Purchasing a Book	Ordered Successfully	Item Purchased and order info sent via mail	Pass
TSD04	Details of users tried to enter in correct database	Entered into database	Database not found	Pass

TSD05	Details of users tried to enter in wrong database	Error message	Invalid Entry	Fail
TSD06	Entered encrypted password to database	Not shown in database	Password is encrypted to 60 character string	Pass
TSD07	Authorized user is logged in	Login successful	User logged in	Pass
TSD08	Authorized admin is logged in	Login successful	Admin logged in	Pass
TSD09	Entered correct Verification Code	Account created and verified	User registered	Pass
TSD10	Entered Wrong Verification Code	Account created but not verified	Not registered, try again	Fail
TSD11	Deleted a User from Database	Successfully Deleted	Deleted a user from the database	Pass
TSD12	Deleted book from Database	Successfully Deleted	Deleted book from the database	Pass

5.2. Front End Testing

Front End Testing is testing Graphical User Interface (GUI), functionality and usability of website or application. The main aim of Frontend testing to make sure that every user is well-protected from bugs. Creating a frontend testing plan helps you to know the devices, browsers, and systems which your project. It involves testing front end integration with database [4].

It includes form validation while taking input into the database through forms. It also ensures that all the required fields in a form are filled appropriately before submitting the form. If the client side is properly validated the result leads to good quality front end interactions.

5.2.1. Test Cases

The test cases identified and the results of each test case is shown in Table 5.2.

Table 5.2: Test Cases for Front End Testing

TESTCASE ID	TESTCASES	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TSFE01	Login as an authenticated user	Login Successful	User Logged in	Pass
TSFE02	Login as an authentic admin	Login Successful	Admin Logged in	Pass
TSFE03	Login with wrong details	Error: Either username or password is wrong!	Invalid Login, Redirected to same page	Pass
TSFE04	Register with correct details	Register Successful	New User added successfully	Pass
TSFE05	Registered with Name as 'Hanan01'	Invalid Registration	Name cannot contain a number!	Pass
TSFE06	Entered an already used username	Invalid Registration	Sorry! Username is already taken	Pass
TSFE07	Enter different password in confirm password	Invalid Registration	Error! Password does not match	Pass
TSFE08	Enter a 4-digit pincode	Invalid Registration	Enter a 6-digit Pincode	Pass
TSFE09	Enter less than 10 digit in contact number	Invalid Registration	Contact number must be of 10 digit	Pass
TSFE10	Leave any of the fields blank	Invalid Registration	Details added to database	Fail
TSFE11	Bookmark without signing in	Not allowed	Login Page is prompted to user	Pass
TSFE12	Buy Book without signing in	Not allowed	Login Page is prompted to user	Pass

TSFE13	Add review without signing in	Not allowed	Login Page is prompted to user	Pass
TSFE14	Buy book after signing in	Direct to transaction page	Successfully directed to transaction page	Pass
TSFE15	Add a book after signing in	Successfully added	Book is Visible on homepage	Pass

5.3. System Testing

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. This stage of testing involves checking security of the system, functionalities of the system being considered. This enumerate the different levels of security available in a system [5].

5.3.1. Test Cases

The test cases identified and the results of each test case is shown in Table 5.3.

Table 5.3: Test Cases for System Testing

TESTCASE ID	TESTCASES	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TSSM01	Admin view visibility to authentic users	Admin page is displayed	Admin site visible after authentication	Pass
TSSM02	Admin view visibility to unauthentic users	Admin Page id displayed	Redirected to login page	Pass
TSSM03	URL access to pages directly without authentication	Blocked	Redirected to login page	Pass
TSSM04	Password encrypted in database	Password stored with encryption	Not visible to even database admin	Pass

TSSM05	Graphs for transaction details month-wise	The graph for specific months should be generated from the google charts	The bar chart is generated showing the transaction details	Pass
TSSM06	Check Bookmark after closing session	The bookmarks will be retrieved as it is from database	The bookmarks are retrieved after again signing in	Pass
TSSM07	Deleting a User by Admin	User will be deleted permanently	Prompt to confirm delete; User is deleted	Pass
TSSM08	Adding a new Category for books	A new category will be added	Category Added!	Pass
TSSM09	Add a Category of same name	New Category will not be created	New category with same name is created	Fail
TSSM10	Sign off from Admin Site	Log off	Sign off successful	Pass
TSSM10	Sign off from User site	Log off	Sign off successful	Pass

Chapter 6: Conclusion

6.1. Limitations

- **Transaction Portal:** The payment gateway has not been provided as that would require the system to integrate with 3rd party APIs which prove to be costlier. However, it may be implemented with any of the available platforms by providing a meager fee for setting up and acquiring the API key.
- **Sorting Option:** The portal does not provide a sorting features for finding the books as per the cost or other attributes.
- **Buying Multiple Books at Same time:** The user cannot buy multiple books at the same time. User must first complete the transaction for one purchase and then the other. Also the quantity of the books is not handled by the portal. It assumes that a seller has only one quantity of uses book. Hence, it will have duplicity of books with different price rates as proposed by the seller.
- **SMTP Mail System:** The current program can send emails using Mercury to local server mails. It requires to purchase a host server connected to internet.
- **Load on database:** The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.

6.2. Future Enhancements

- **Add Payment Gateway:** Adding a payment portal for making online transactions to make purchasing of books more convenient to users.
- **Enquiry about Books:** The user can ask enquiry about the status of a particular book directly to the seller. A personal chat feature will meet this requirement
- **Discussion Platform:** Provide a free and open discussion platform where users from various colleges can share their experiences and guide the juniors in their career.
- **PDF E-Books:** The portal should provide some e-books for free of cost for the convenience of students.

References

- [1] "Javatpoint," 2018. [Online]. Available: <https://www.javatpoint.com/dbms-er-model-concept>.
- [2] "UML-Diagrams," 2015. [Online]. Available: <https://www.uml-diagrams.org/>.
- [3] "GeeksForGeeks," 2018. [Online]. Available: <https://www.geeksforgeeks.org/software-testing-database-testing/>.
- [4] "FrontEnd Masters," 2018. [Online]. Available: <https://frontendmasters.com/books/front-end-handbook/2018/tools/testing.html>.
- [5] "Guru99," 2019. [Online]. Available: <https://www.guru99.com/system-testing.html>.
- [6] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Pearson, 2017.
- [7] M. Rathi, S. Badjate and P. Kashirsagar, "Web Portal for Selling and Purchasing Old Books," *International Research Journal of Engineering and Technology (IRJET)*, vol. IV, no. 6, pp. 3330-3332, 2017.
- [8] Q. L. Chen and Y. H. Feng, "The design and application of the online bookstore system .Net based on three tier architecture," in *10th International Conference on Information, Communications and Signal Processing (ICICS)*, Singapore, 2015.

APPENDIX A: Code Snippets

```

<?php
function customPageHeader(){
    echo "<link rel='stylesheet' type='text/css' href='includes/css/main_page.css'>";
    echo "<script type='text/javascript' src='includes/javascript/main_page.js'></script>";
}

?>
<?php include "./templates/header.php"; ?>
<style type="text/css">
    .viewMore {
        height: 300px;
        background-color: #d5e6f3
    }
    .books {
        padding-right: 15px;
        padding-left: 5px;
    }
</style>

<?php
if (isset($_SESSION['username'])) {
    $username = $_SESSION['username'];
    $verify_query = "SELECT * FROM users WHERE username = '{$username}'";

    $verify_query_result = mysqli_query($connection,$verify_query);

    if(!$verify_query_result){
        die("FAILED" . mysqli_error($connection));
    }else{
        $verify_row = mysqli_fetch_assoc($verify_query_result);
        $email_id = $verify_row['email'];
        $username = $verify_row['username'];
        $is_verified = $verify_row['is_verified'];

        if ($is_verified == 'false') {
            header("Location: verify_user.php?email_id='{$email_id}'&username='{$username}'");
        }
    }
}
?>

```

```

<div id="wrapper">
<?php include "./templates/navigation.php"; ?>
<div class= "container close_bookmark_sidebar" id="page-content-wrapper" style="margin: 0px; padding: 0px">
<div class="container-fluid row" style=" background-image:url('includes/images/image.jpg'); background-repeat: no-repeat
; background-size:100% 100%; height: 400px;">

<br>
<br>

<!--<p align="center" style="color: #484848; font-size: 30px;font-family: fantasy;">Buy and Sell Books Online</p-->
<head><link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap" rel="stylesheet"></head>
<p align="center" style="background-color: rgb(0,0,0); background-color: rgba(0,0,0, 0.7); color: #FFF; font-size: 30px
;font-family: 'Open Sans', sans-serif; font-weight: 900;">RVCE BUY AND SELL BOOKS ONLINE</p>
<!-- search box start -->
<form method="GET" action="./search.php">
    <div class="container-fluid close_bookmark_sidebar autocomplete" style="background-color: transparent; margin-top:
20px;">
        <div class="full-width-util input-group" id="searchBox">
            <input id="myInput" type="text" name="search" class="form-control home-search-bar acInput" placeholder="Search by Book title..." autocomplete="off" style="width:100%; background-color:#fff" required>
            <span class="input-group-btn">
                <button type="submit" value="Search" id="searchButtonInline" class="btn btn-primary no-top-margin">Go!</
button>
            </span>
        </div>
        <div id="parent_element" style="background-color:white;"></div>
    </div>
</form>
<!-- search box end -->

<script>

    autocomplete(document.getElementById("myInput"),document.getElementById("parent_element"));

</script>

```



```

</div>
<br>
<div class="container close_bookmark_sidebar" id='container'>
    <div class="row" >
        <!-- id="#category"-->
        <div class="col-sm-3">
            <h2>Categories</h2>
            <?php include "templates/sidebar.php"; ?>
        </div>
        <div class="col-sm-9">

<?php
    $query = "SELECT * FROM categories WHERE parent_category_id = 0";
    $category_result = mysqli_query($connection,$query);

    if(!$category_result){
        die("QUERY FAILED ".mysqli_error($connection));
    }else{
        while($category_row = mysqli_fetch_assoc($category_result)){
            $category_id = $category_row['category_id'];
            $category_name = $category_row['category_name'];

            //THIS ARRAY WILL CONTAIN ALL CATEGORIES AND SUB CATEGORY IDS FOR A PARTICULAR CATEGORY
            $all_category_ids = array();
            $all_category_ids[] = $category_id;

```

```

$sub_query = "SELECT * FROM categories WHERE parent_category_id = {$category_id}";

$sub_query_result = mysqli_query($connection,$sub_query);

if(!$sub_query_result){
    die("QUERY FAILED ".mysqli_error($connection));
} else{
    while($sub_query_row = mysqli_fetch_assoc($sub_query_result)){
        $sub_category_id = $sub_query_row['category_id'];
        $all_category_ids[] = $sub_category_id;
    }
}

$title_printed =false;

foreach ($all_category_ids as $single_category_id) {
    $product_query = "SELECT * FROM books WHERE category_id = {$single_category_id} AND book_status='available' ORDER BY date DESC";

    $product_query_result = mysqli_query($connection,$product_query);

    if(!$product_query_result){
        die("QUERY FAILED ".mysqli_error($connection));
    } else{
        $num_products = mysqli_num_rows($product_query_result);
        if($num_products != 0){
            if(!$title_printed){
                $count = 0;
                echo "<h3>{$category_name}</h3>";
                //row div started
                echo "<div class='row'>";
                $title_printed = true;

                while($count < 4 && $product_row = mysqli_fetch_assoc($product_query_result)){
                    $count = $count + 1;
                    $book_id = $product_row['book_id'];
                    $book_name = $product_row['book_name'];
                    $book_author = $product_row['author'];
                    $book_edition = $product_row['edition'];
                    $book_subject = $product_row['subject'];
                    $book_price = $product_row['book_price'];
                    $book_image = $product_row['book_image'];

                    <div class="books col-sm-6 col-md-3 col-lg-3 col-xs-6">
                        <div class="thumbnail">
                            <div class="w3-display-container w3-hover-opacity">
                                " style="width:100%; height: 230px;">
                            <div class="w3-display-middle w3-display-hover">
                                <a href="book_details.php?book_id=<?php echo $book_id?>">
                                    <button class="w3-button" style="background-color: #0b113e; color: white">View Details</button></a>
                            </div>
                        </div>
                    </div>
                }
            }
        }
    }
}

```

```

        <div class="w3-display-topright w3-display-hover">
            <a onclick="showQuickView('<?php echo $book_name; ?>','<?php echo $book_author; ?>','<?php echo $book_price; ?>','<?php echo $book_subject; ?>','<?php echo $book_image; ?>','<?php echo $book_id; ?>');" data-toggle="modal" href="" data-target="" style="color: white" title="quick look"><button><span class="glyphicon glyphicon-zoom-in" style="color: black"></span></button></a>
        </div>
        <div class="caption" style="height: 60px; border-top: 5px solid blue">
            <p align="center"><?php echo $book_name ?></p>
        </div>
    </div>
</div>
}
}

if($title_printed){
    //row div end
    echo "</div>";
}

}
}

?>
</div>
</div>
</div>

<div id="quick_view" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <span><h4 style="width: auto;">Quick Look</h4>
                <button type="button" class="close" data-dismiss="modal" style="margin-right: 9px; width: auto;">&times;
                </span>
            </div>
            <div class="modal-body">
                <div class="row">
                    <div class="col-sm-5">
                        <img style="max-width:300px;max-height:300px;" id="modal_image" src="">
                    </div>
                    <div class="col-sm-7">
                        <div id="modal_name" style="font-size:28px; font-weight:500;"></div>
                        by <span id="modal_author" style="font-size:20px; font-weight:400;"></span><br><br>
                        Subject <span id="modal_subject" style="font-size:18px; font-weight:400;"></span><br><br>
                        &#8377<span id="modal_price" style="font-size:20px; font-weight:400;"></span><br><br>
                        <a id="modal_view_more" class="btn btn-primary">View More</a>
                    </div>
                </div>
                <div class="modal-footer" style="margin-right: 10px;">
                    <button type="button" class="btn btn-default" data-dismiss="modal" style="position: absolute; margin-bottom: 10px; margin-right: 10px; right: 0; bottom: 0">Close</button>
                </div>
            </div>
        </div>
    </div>
</div>
<?php include "templates/footer.php"; ?>

```

Figure 0.1: Code Snippet-1 "index.php" for homepage

```

<?php
function customPageHeader(){
    echo "<link rel='stylesheet' type='text/css' href='includes/css/main_page.css'>";
    echo '<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">';
    echo "<script type='text/javascript' src='includes/javascript/main_page.js'></script>";
}

?>
<?php
if(isset($_GET["search"])){
    $search_index = $_GET["search"];
}
?>

<?php include "./templates/header.php"; ?>
<div id="wrapper">
<?php include "./templates/navigation.php"; ?>
<div class= "container close_bookmark_sidebar" id="page-content-wrapper" style="margin: 0px; padding: 0px">

<p align="center" style="color: white; font-size: 30px;font-family: fantasy;">Buy and Sell Books Online</p>
</div>
<br>
<div class="container close_bookmark_sidebar" id='container'>
    <div class="row" >
        <!-- id="#category"-->
        <div class="col-sm-3">
            <h2>Categories</h2>
            <?php include "templates/sidebar.php"; ?>
        </div>
        <div class="col-sm-9">
            <h2>Search Result for <u><?php echo $search_index; ?></u></h2>
        <br>
        <div class = "row">

<?php
$search_index = mysqli_real_escape_string($connection,$search_index);
$query = "SELECT * from books WHERE book_name LIKE '%$search_index%'";
$query_result = mysqli_query($connection,$query);

if(!$query_result){
    die("QUERY FAILED ".mysqli_error($connection));
} else{
    if (mysqli_num_rows($query_result) == 0) {
        echo "<h3 style='padding-left:17px;'>No Book found</h3>";
    }
    while($product_row = mysqli_fetch_assoc($query_result)){
        $book_id = $product_row['book_id'];
        $book_name = $product_row['book_name'];
        $book_author = $product_row['author'];
        $book_edition = $product_row['edition'];
        $book_subject = $product_row['subject'];
        $book_price = $product_row['book_price'];
        $book_image = $product_row['book_image'];
    }
}
?>
        <div class="col-sm-6 col-md-3 col-lg-3 col-xs-6">
            <div class="thumbnail">
                <div class="w3-display-container w3-hover-opacity">
                    " style="width:100%; height: 230px;">
                    <div class="w3-display-middle w3-display-hover">
                        <a href="book_details.php?book_id=<?php echo $book_id?>">
                            <button class="w3-button" style="background-color: #0b113e; color: white">View Details</button></a>
                    </div>
            </div>
        </div>
    </div>
</div>

```

```
<script type="text/javascript">
    function f1() {
        document.getElementById('modal01').style.display='block';
    }
</script>
<button class="glyphicon glyphicon-zoom-in" style="color: black; size: 32px;" onclick="f1()"></button>
</div>
</div>
<div class="caption" style="height: 60px; border-top: 5px solid blue">
    <p align="center"><?php echo $book_name ?></p>
</div>
</div>
<?php
    }
}
?>
</div>

</div>
</div>
</div>
<div id="modal01" class="w3-modal" onclick="this.style.display='none'">
    <span class="w3-button w3-hover-red w3-xlarge w3-display-topright">&times;</span>
    <div class="w3-modal-content w3-animate-zoom">
        
    </div>
</div>
</div>
<?php include "templates/footer.php"; ?>
```

Figure 0.2: Code Snippet - "search.php" for homepage

APPENDIX B: Screenshots

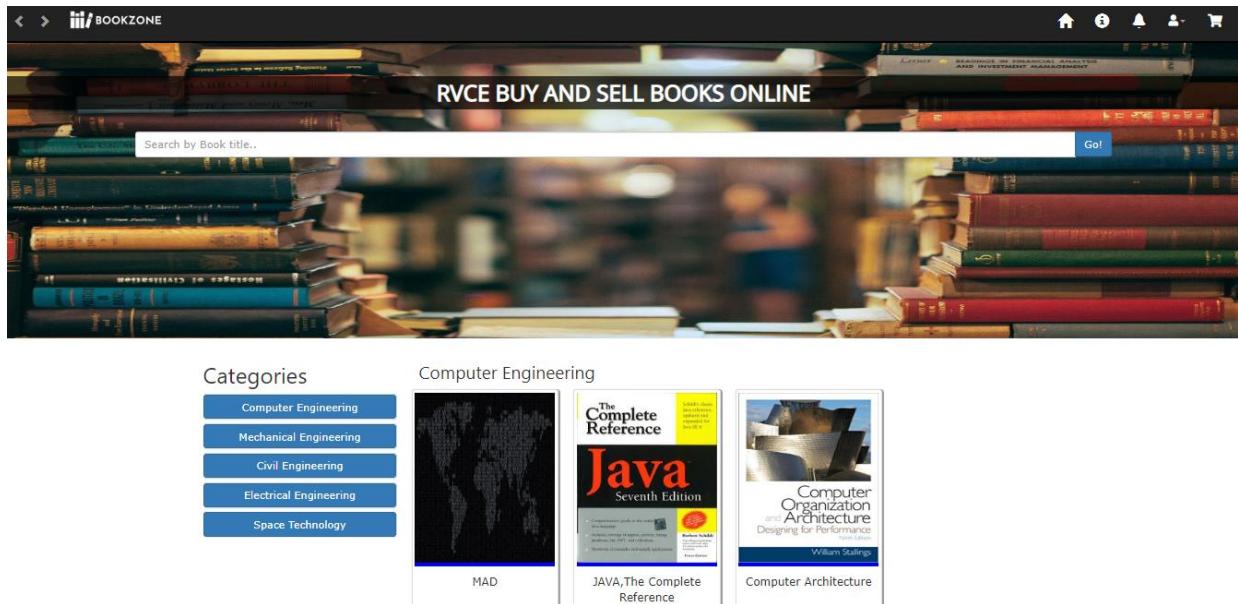


Figure 0.1: Homepage

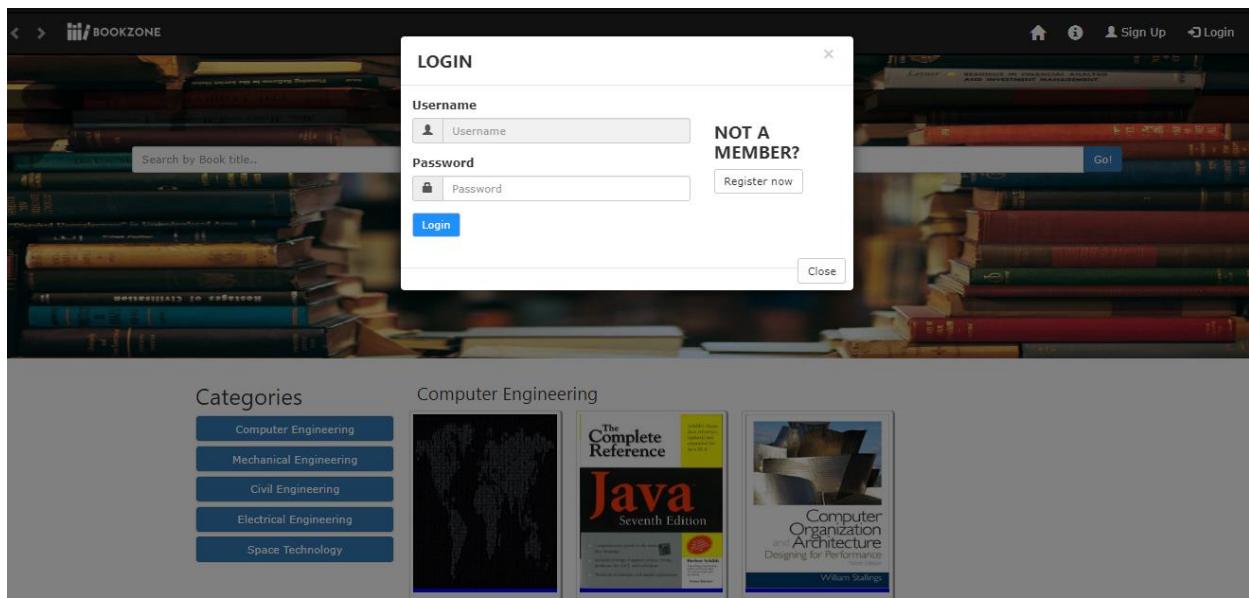


Figure 0.2: Login Form

Registration Form

Username

Password

Confirm Password

Email

First Name Middle Name Last Name

City Pincode

Contact Your Preference

Verify and Register

Figure 0.3: User Registration Form

Data Structures Using C
by Reema Thareja

Was : ₹300
Now : ₹ 150 50% off

★ ★ ★ ★ Write a review See all reviews

Buy Now **Bookmark**

Subject: DS
Category: Electrical Engineering
[More Details](#) [More from this Seller](#)

DESCRIPTION	DETAILED INFO
<p>Data Structures Using C has been developed to provide a comprehensive and consistent coverage of both the abstract concepts of data structures as well as the implementation of these concepts using C language. It begins with a thorough overview of the concepts of C programming followed by introduction of different data structures and methods to analyse the complexity of different algorithms. It then connects these concepts and applies them to the study of various data structures such as arrays, strings, linked lists, stacks, queues, trees, heaps, and graphs.</p>	<p>Subject: DS Category: Electrical Engineering Book Name: Data Structures Using C Author: Reema Thareja Original Price: 300</p>

Figure 0.4: View Book Details

More from other Sellers

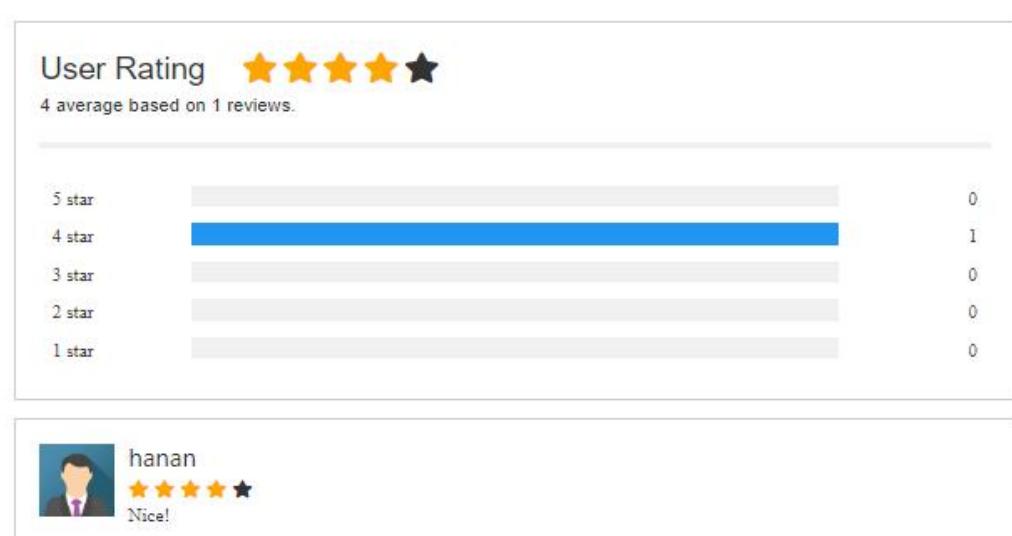
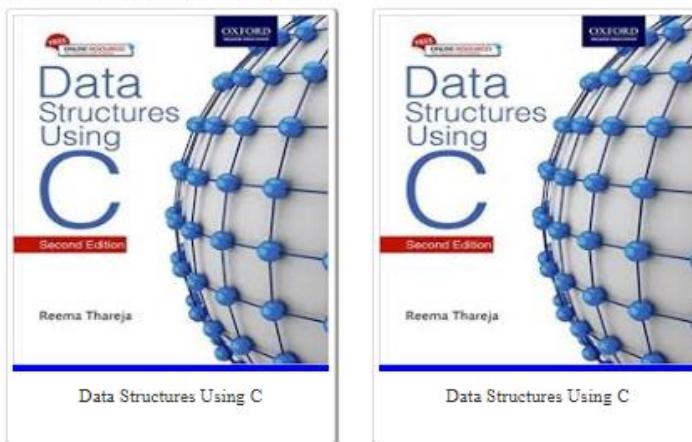


Figure 0.5: Review Page

The image shows a screenshot of a web page for a book titled 'Data Structures Using C' by Reema Thareja, Second Edition. On the left, there is a thumbnail of the book cover. In the center, there is a 'User Rating' section with a 4.5-star average based on 1 review. On the right, a modal window titled 'Write a review' is open. The modal contains fields for 'Ratings' (set to 5), 'Comments' (containing the text 'Goood!!'), and a 'Submit' button. There is also a 'Close' button in the top right corner of the modal.

Figure 0.6: Form for adding a review

Tell us more about your book

Book Name <input type="text" value="Enter book name"/>	Author <input type="text" value="Enter author name"/>
Selling Price <input type="text" value="₹ Enter Selling price"/>	Original Price <input type="text" value="₹ Enter Original price"/>
Subject <input type="text" value="₹ Enter Subject"/>	Category <input type="text" value="Choose..."/>
Edition <input type="text" value="Enter book edition"/>	Post Image <input type="button" value="Choose File"/> No file chosen
Book Description <div style="border: 1px solid #ccc; padding: 5px; height: 50px;"> Paragraph B I ♂ ♀ = ≡ “” ↔ ↶ ↷ </div>	
<input type="button" value="Save Details"/>	

Figure 0.7: Form for Selling a Book



Figure 0.8: Admin Homepage

The screenshot shows the 'ALL REVIEWS' section of the admin interface. On the left is a sidebar with links: Dashboard, Recent activities, Users, View All Books, Categories, and Reviews. The main area has a title 'ALL REVIEWS' and a search bar. It displays two entries in a table:

BookId	Username	BookName	Ratings	Review	Edit Review	Delete
18	hanan	Data Structures Using C	4	Nicel	<button>Edit Review</button>	<button>Delete</button>
22	hanan	Theory of Automata	5	Nicee	<button>Edit Review</button>	<button>Delete</button>

Showing 1 to 2 of 2 entries

Figure 0.9: Admin view of All Reviews

The screenshot shows the 'ALL BOOKS' section of the admin interface. On the left is a sidebar with links: Dashboard, Recent activities, Users, View All Books, Categories, and Reviews. The main area has a title 'ALL BOOKS' and a search bar. It displays a list of books in a table:

SellerName	BookName	Author	Edition	Subject	Price	OriginalPrice	View More	Edit	Delete
hanan	MAD	Hanan	1	Android	250	500	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
harsh	Computer Architecture	William Stalling	6	COA	450	650	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
harsh	JAVA,The Complete Reference	Herbert Schildt	5	OOPM	499	700	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
harsh	Fundamental of Fluid Mechanics	Rothmayour	7	Fundamental of Fluid Mechanics	130	250	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
harsh	Theory of Automata	Rajeev Motwani	3	Theory of Automata	350	450	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
harsh	Mechanical Engineering Design	Kiess Nisbett	10	Mechanical Engineering Design	89	195	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
murtaza	Kumbhojkar	Kumbhojkar	6	Maths	250	400	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
murtaza	JAVA,The Complete Reference	Herbert Schildt	5	OOPM	550	700	<button>View More</button>	<button>Edit</button>	<button>Delete</button>
murtaza	Computer Architecture	William Stalling	6	COA	550	650	<button>View More</button>	<button>Edit</button>	<button>Delete</button>

Figure 0.10: Admin View of All Books