BOOK BAZAAR: ONLINE BOOK RESELLING AND DONATING

A MINI PROJECT REPORT

Submitted by

ARUL MARIA AGNES I - (6176AC21UCS009)

CHANDRU C - (6176AC21UCS014)

DHANRAJ P - (6176AC20UCS023)

KABILAN M - (6176AC20UCS055)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

DR.M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY: : CHENNAI 600 025

NOVEMBER 2024

BOOK BAZAAR: ONLINE BOOK RESELLING AND DONATING

A MINI PROJECT REPORT

Submitted by

ARUL MARIA AGNES I - (6176AC21UCS009)

CHANDRU C - (6176AC21UCS014)

DHANRAJ P - (6176AC20UCS023)

KABILAN M - (6176AC20UCS055)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

DR.M.G.R NAGAR, HOSUR-635130

ANNA UNIVERSITY: : CHENNAI 600 025

NOVEMBER 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this mini project report **"BOOK BAZAAR: ONLINE BOOK RESELLING AND DONATING"** is the Bonafide work of **"ARUL MARIA AGNES I (6176AC21UCS009), CHANDRU C (6176AC21UCS014), DHANRAJ P (6176AC21UCS023), KABILAN M (6176AC21UCS055)"** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. G. FATHIMA, M.E., Ph.D.,	**MRS. M. KASTHURI, M.E., (Ph.D.,)**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR,	ASSISTANT PROFESSOR,
Department of CSE,	Department of CSE,
Adhiyamaan College of Engineering, (Autonomous), Dr. M.G.R. Nagar,	Adhiyamaan College of Engineering, (Autonomous), Dr. M.G.R. Nagar,
Hosur – 635 130.	Hosur – 635 130.

Submitted for the Mini project VIVA-VOCE Examination held on _____at **Adhiyamaan College of Engineering (Autonomous), Hosur-635 130**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Expressing profound gratitude, I wish to acknowledge the invaluable support and guidance I've received during the execution of my mini project at Adhiyamaan College of Engineering.

We show our sincere thanks to **Dr. R. RADHAKRISHNAN, M.E., Ph.D.,** Principal, Adhiyamaan College of Engineering Hosur, for this valuable opportunity and environment given to us.

We extend our hearty gratitude to **Dr. G. FATHIMA, M.E., Ph.D.,** Professor and Head of the Department, Department of Computer Science and Engineering, Adhiyamaan College of Engineering Hosur, for her guidance and valuable suggestions and encouragement throughout this project.

We are highly indebted to our Supervisor **Mrs. M. KASTHURI, M.E., (Ph.D.,),** Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, whose immense support, encouragement and valuable guidance were responsible for completing the project successfully.

We extend our hearty thankfulness to our project coordinator and all our department faculty for their support in helping us complete this project successfully.

This project would not have been possible without the collective support and guidance of these individuals and the Adhiyamaan College of Engineering community.

Thank you for making this endeavor a success.

ABSTRACT

This project, titled "Online Book Reselling and Donating," was developed to streamline the buying, donating, and reselling of books through an accessible web-based platform. The Book bazaar providing options for donation and resale to support a sustainable cycle of book usage. Additionally, an online payment gateway is integrated, ensuring safe and straightforward transactions for both book purchases and resales. The administrator has comprehensive control over the platform, including access to customer information, payment records, inventory updates, and purchase history. This role allows the admin to oversee the book inventory, add new books, manage donations, and track resales. In contrast, the customer can register their details online, access their profile, view personal purchase history, and engage in transactions while remaining restricted from administrative functions. The system's secure login authentication ensures that users' data is protected and that access remains role-specific, preventing customers from viewing or editing administrative details. Overall, this project aims to foster a sustainable and user-friendly platform for academic book sharing, enabling efficient communication and management between users and administrators. To maintain security and role-specific access, the platform employs a secure login authentication system that strictly controls who can access each level of information. This system protects customer data from unauthorized access, ensuring that only verified users with appropriate permissions can view or modify specific information. Administrators can confidently manage all aspects of the platform, while customers enjoy a personalized, private user experience. By fostering this secure, well-organized environment, the project aims to create a sustainable, user-friendly online community where academic book reselling and donations are facilitated with ease, bridging communication and trust between users and administrators in a safe, efficient manner.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

HTML	-	**H**yper **T**ext **M**arkup **L**anguage
CSS	-	**C**ascading **S**tyle **S**heet
JS	-	**J**avaScript
PHP	-	**H**ypertext **P**reprocessor
SQL	-	**S**tructured **Q**uery **L**anguage
ACE	-	**A**dhiyamaan **C**ollege of **E**ngineering

CHAPTER I

INTRODUCTION

This project entitled "**Online Book Reselling and Donating**" was developed using PHP as a Front End and MY SQL as Back End. The project "Data Collections in Library" is developed for customizing library activities through or over the internet. This project handles several parts. There are 1. Online Purchasing, 2. Online Donation 3. Reselling. This project is also based on Library and Communication system. In the management Part, the administrator has all the access rights to the system. The admin login to the system to view the details, View payment details, and adds the new available Book into the database. The customer details are also maintained by the administrator. The book purchase details and return details are also maintained. From the Student's point of view, the customer registers their details online. After registration is completed he/she views the available books in the library and studies the book online. Here the purchase details are also viewed by the Customer and Admin. Pay the amount through an online payment. Here the administrator and Customer have their own login authentication.

1.1 PROJECT OVERVIEW:

Admin

- New Book Entry and Reselling
- Add Donate Books
- View Customer Details
- View Order
- Update Status

User

- Registration
- View Book Details
- View Donate Books
- Place Order
- View Status

1.2 PROJECT OBJECTIVE:

The main objective of the project is to create an online book reselling and donation store that allows users to search and buy a book online based on title, author, and subject. Users which select books are displayed and the user can purchase their books online. Using this website, users can purchase a book, resell and donate online instead of going out to a bookstore and wasting time. The website is focused on the donation section which is the most important part. An Online Book store is an online web application where the customer can buy, sell and donate books online. Through a web browser the user can search for a book by its title or author, later can add it to the shopping cart, and finally purchase books online.

1.3 PROJECT DEFINITION:

To build a website for reselling books. The platform allows the user to buy, and resell as per they wish. There is an extra option included here called donate which allows the user to donate the books to some NGOs. Here the user can view the product, add them to the cart, and buy the product. Thus, this website is time-saving and easy to access for the user.

Many NGOs are existing that run programs like Donate–A–Book. But under the program, they collect money and with the money, books are procured. But there should be some mechanism that will make use of the available books rather than procuring new books as a part of helping needy people.

1.4 MODULES DESCRIPTION

- **Authentication**

This module contains all the information about the authenticated user. An administrator without his username and password can’t enter into the login if Admin is only the authenticated user then he can enter his login. Authentication is the process of verifying the identity of a user by obtaining some sort of credentials and using those credentials to verify the user's identity.

- **Add Book**

In this Module, the Administrator Adds the Book Information. It Contains Information about the Book id, Book name, Author name, publisher, and year. The administrator can finally store the information in the database. Administrator Maintain the Book Details.

- **View Book Buy order**

In this Module, the Administrator Views the Book order Information for the User. It Contains Information about the book id, user id, Book id, Book name, Author name, publisher, and year. The administrator only has to permission book the requested Details.

- **User Login**

This module contains all the information about the authenticated Student. Authentication is the process of verifying the identity of a user by obtaining some sort of credentials and using those credentials to verify the user's identity. If the credentials are valid, the authorization process starts.

- **User Registration / Login**

Users who want to buy Products should register with the website by providing personal details. After the registration, he will be issued with valid user id and password by the Administrator. After successfully login into the system, the user moves to the instruction web page where he will get instructions about the process.

- **View Status**

In this Module, Customer view the Product Delivery Status from the Administrators.

- **Donate Book**

In this module, the User Donates the Book Information. It Contains Information about the user id, Book id, Book name, Author name, publisher, and year.

CHAPTER 2

LITERATURE SURVEY

Ansari Sharim Iqbal Ansari Sharique Taj. Naik Owais Anwar, Mr. Kalidas Bhawale et al. [1] , Gyanchand Dhiwar Kaushik Avinash et al. [2] , Riya Kalburgi, Mohammed Faiz Sayyed et al. [3] , PAUL KIPLING T.D. WILSON et al. [4] , Trish Groves et al. [5] . All of which are centered around the themes of e-learning acceptance, adoption, and effectiveness. These studies collectively delve into the complexities of e-learning in the context of education.

Ansari Sharim Iqbal Ansari Sharique Taj. Naik Owais Anwar, Mr. Kalidas Bhawale et al. [1] this paper discusses the development of an e-commerce website that facilitates book buying, reselling, and donations. It explores various methods for connecting buyers and sellers and presents the architecture of the platform.

Gyanchand Dhiwar Kaushik Avinash et al. [2] Focuses on automating the inventory management system of an online bookstore that specializes in secondhand books, enhancing user experience and operational efficiency.

Riya Kalburgi, Mohammed Faiz Sayyed et al. [3] Provides access to a platform dedicated to donating books within a community. The system was designed with an easy-to- use interface to encourage community engagement and literacy growth.

PAUL KIPLING T.D. WILSON et al. [4] This paper investigates the ways in which publishing and bookselling industries have adapted to the internet, including the creation of reselling platforms.

Trish Groves et al. [5] It highlights the role of organizations that support book donations and provides information on the types of books and journals that are in demand.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing system Online book store System is conducted manually. The User who needs to request for book needs to go to the librarian and he has to refer to the types of books available in the library it should be a manual process. The book details, member's details, request details, notice board details, etc. are maintained manually.

The current Online Bookstore System operates entirely manually, meaning that users who wish to request a book must physically visit the store or contact the administrator directly. In order to find out what books are available, users need to consult the administrator, who must manually check the inventory or records. This process not only consumes time for the user and the administrator but also lacks the convenience and efficiency users expect from an online bookstore. Details of book inventory, customer information, book requests, and other notices are all recorded and managed manually, leading to inefficiencies and a higher risk of errors.

One major disadvantage of this manual system is its reliance on extensive manpower to record and track information about every customer and transaction. Every new customer has to be individually documented by the administrator, who must also update customer records for each visit. This requirement for continuous data entry and record-keeping can become overwhelming, especially as the store's customer base grows. The administrator must also manage book requests, keeping track of orders, returns, and other transactions without the support of an automated system, which increases the workload and the potential for inaccuracies.

From the customer's perspective, this manual system is less convenient and introduces more uncertainty into the book selection process. Since book availability must be verified manually by the administrator, customers cannot simply browse available titles online and must instead depend on the

administrator for information. This lack of transparency makes it difficult for customers to know whether their desired books are currently in stock, increasing the likelihood of disappointment or frustration. Customers also face the inconvenience of re-registering their information each time they make a purchase, as the system lacks an automated user profile feature to store returning customers' details.

The manual process also complicates inventory management for the store administrator, who must track book availability and handle multiple customer inquiries without an automated system. Without real-time inventory tracking, it becomes difficult to maintain an accurate picture of stock levels, risking stockouts and overordering. The system is particularly strained when multiple customers need assistance simultaneously, as the administrator may struggle to keep up with the demand. Ultimately, this manual approach limits the efficiency and scalability of the online bookstore, making it challenging to meet the needs of a growing user base while maintaining accurate and up-to-date records.

Disadvantages

- It needs manpower to record all the details of all customers
- On client's view, it's a riskier job to view the availability of books manually
- Each and every time the student's details have to be registered if the student often visits the library
- Checking out the availability of stocks is a risk process for the administrator of the store.
- On the administrator side, it's difficult to manage more than no user if visited.

3.2 PROPOSED SYSTEM

The proposed system should overcome all the disadvantages of the existing system. The existing system is not functioning well due to the manual process. Thus the proposed system should minimize the manual efforts and fully computerized online process. In such a way that it should support the User

request for book details sent to the Administrator. And also the administrator has to keep track of all the User registered and requested book details. Time consumption for arrangement will be minimum. The customer enters their login id for request and downloads for the book.

The proposed system aims to address the limitations of the existing manual Online Bookstore System by introducing a fully computerized, automated platform for book requests and inventory management. By transitioning to an online system, the new solution will significantly reduce the manual effort required from both users and administrators. Users will be able to access book details, check availability, and make requests online without needing to contact the administrator directly. This will not only streamline the request process but also ensure that users can conveniently access real-time information on the available books, improving their overall experience. Meanwhile, the administrator will be able to handle requests and track user information seamlessly, reducing the time and effort previously required to manage these tasks manually.

The proposed system will provide a secure login portal where each user, including both new and returning customers, can create a unique profile. Users will be required to enter a valid username and password to access the system, ensuring secure and individualized access. Once logged in, users can easily browse the available books, request specific titles, and even download digital copies, if available. This automated process will enhance efficiency and save time for the user, who can independently view and request books without needing assistance from the administrator. Additionally, users' data, such as past transactions and book requests, will be securely stored in their profiles, making it easier for them to track their order history and for the administrator to access relevant user information.

From the administrator's perspective, the proposed system will allow for easy access to all stored information, including registered users, requested books, and current inventory levels. This centralized database will replace the manual record-keeping system, providing accurate, real-time data on all library

transactions and resources. With an automated, searchable database, the administrator can quickly retrieve and update information, thereby reducing time spent on routine tasks and minimizing errors. By automating the book request and tracking process, the proposed system will ensure that library data can be accessed and managed effectively, eliminating the need for repetitive manual work and supporting more efficient bookstore operations.

In addition to streamlining user and administrator interactions, the proposed system will also incorporate automated notifications and reminders, further enhancing the efficiency of the online bookstore. Users will receive updates regarding the status of their requests, including notifications when a requested book is available for download or pick-up. Administrators, in turn, will benefit from automated alerts regarding low inventory levels or pending requests, enabling proactive management of resources. These notifications reduce the need for manual follow-ups and ensure that users and administrators are kept informed at each step of the process. By centralizing and automating communication within the system, the proposed solution not only improves the user experience but also allows administrators to allocate their time and resources more effectively, ultimately supporting a smoother and more organized bookstore operation.

Advantages

- The User can enter only if the username and the password are correct.
- Time-Saving.
- The Administrator all saved information can be viewed.
- All the library data can be accessed easily whenever needed and so the manual work can be reduced.

3.3. PROPOSED SOLUTION

The proposed solution for the online bookstore aims to address the limitations of the current manual system by implementing a fully automated, user-friendly digital platform. This new system will streamline interactions

between users and the administrator, reduce the need for repetitive manual data entry, and provide a smooth, convenient experience for users seeking book information, requests, or downloads. By digitizing key processes, this system will help both users and administrators save time, maintain accurate records, and ensure a secure and seamless book request and fulfilment process.

The solution will feature a secure user authentication system, where each user must log in with a unique username and password to access the bookstore. This login process ensures that only registered users can make requests or download books, thus safeguarding user information and maintaining system integrity. Upon logging in, users will have access to a personalized dashboard that displays their past transactions, current requests, and any available download links. This personalized access helps users easily manage their book requests and activities without needing additional support from the administrator.

In addition to providing a streamlined user experience, the proposed system will optimize time management for both users and the administrator. Users will have direct access to a digital catalogue that shows real-time book availability, removing the need for manual confirmation from the administrator. With the catalogue and inventory digitally organized, users can quickly browse, request, and download books at their convenience. This efficient self-service feature eliminates waiting times, offering users 24/7 access to book information and reducing the administrator's workload.

For the administrator, the system centralizes all user, book, and request data into a single, organized interface. This allows the administrator to view and manage data easily, track book inventory in real-time, and monitor user requests without needing to sort through physical records. With this centralized data view, the administrator can process requests, update records, and respond to user inquiries quickly, thereby improving the overall response time and reducing the likelihood of errors. Automated data tracking also means that the administrator can handle a larger volume of requests and users with minimal additional effort.

The proposed solution enhances data security, accuracy, and accessibility, all of which contribute to a more reliable system. By storing all records digitally, the system minimizes the risk of data loss and ensures that information is easily retrievable whenever needed. Additionally, the automation of data organization and management reduces the chances of human error, leading to more accurate records and a smoother operational flow. With sensitive user and transaction data stored securely, the system further protects privacy and ensures that only authorized users can access confidential information.

Overall, this proposed system will provide a robust and efficient foundation for the online bookstore, allowing it to meet the demands of a growing user base while improving the user experience and reducing administrative workload. By modernizing key processes, the solution will help the bookstore function smoothly, saving time for users and administrators alike, while maintaining secure, accurate, and easily accessible records. This comprehensive approach will create a scalable, effective platform for both current operations and future growth.

3.4. PROBLEM SOLUTION FIT

Creating an effective online platform for book reselling and donating requires a solid foundation that addresses customer needs, streamlines book management, and promotes seamless transactions. Here's a complete problem-solution kit that could help in building a user-friendly, scalable, and community-driven online book reselling and donation platform:

1. Lack of Platform for Book Reselling and Donation:

Problem: How can we create a specialized online platform that effectively serves the needs of users looking to resell or donate books?

Solution: To meet the needs of users interested in reselling or donating books, develop a comprehensive website and mobile app. This platform should cater specifically to buying, selling, and donating books, providing dedicated sections for each function. By offering a centralized hub for these activities, users can

easily participate in transactions and donations, making the platform a go-to resource for book exchanges.

2. User Accessibility and Experience:

Problem: How can we ensure that users find the platform easy to navigate and visually appealing across different devices?

Solution: An intuitive and visually engaging user interface (UI) can be achieved by incorporating a clear layout with a search bar, filters by genre, condition, and language, and categorizing book listings effectively. The platform should be fully responsive, ensuring smooth navigation on both web and mobile, which will help users locate books or post listings efficiently.

3. Inventory Management for Sellers and Donors:

Problem: How can sellers and donors manage their book inventories effectively and track their listings?

Solution: Introduce an inventory management system within the platform where users can easily list books, add details like title, author, ISBN, condition, and price, and track their stock. This system enables sellers to update availability, maintain control over inventory, and manage donations in real-time, which will streamline their experience and provide better service to buyers and recipients.

4. Quality Control of Listed Books:

Problem: How can we maintain quality control for listed books, ensuring buyers receive what they expect?

Solution: Implement a book condition rating system to maintain consistent quality standards, with categories like "new," "good," and "acceptable" for clarity. Sellers can upload photos and provide additional descriptions, while an ISBN autofill feature ensures accurate listing details. This transparent rating and image requirement fosters trust among buyers and sets quality expectations for transactions.

5. Streamlining the Selling and Donating Process:

Problem: How can we streamline the process of selling and donating books to make it as convenient as possible for users?

Solution: Simplify transactions by offering a one-click selling option where a barcode scanner autofill's essential information, reducing manual entry. For donations, arrange pickup options with local partners or provide drop-off locations, allowing donors to contribute without hassle and ensuring recipients can easily access donated books.

6. Building Trust and Community Among Users:

Problem: How can we build a trustworthy community on the platform, promoting positive interactions between users?

Solution: Establish a review and rating system to encourage honest feedback on book quality and transaction experiences, helping to build trust within the community. Detailed user profiles showcasing transaction history will promote transparency, while positive interactions increase user confidence and platform credibility.

7. Lack of Visibility and Outreach:

Problem: How can we attract users to the platform and increase its visibility online?

Solution: Drive traffic and increase visibility through targeted digital marketing, including SEO to capture search engine traffic, social media marketing to showcase books and platform features, and email campaigns highlighting new arrivals or ongoing donation drives. These efforts will boost awareness and attract diverse users to the platform.

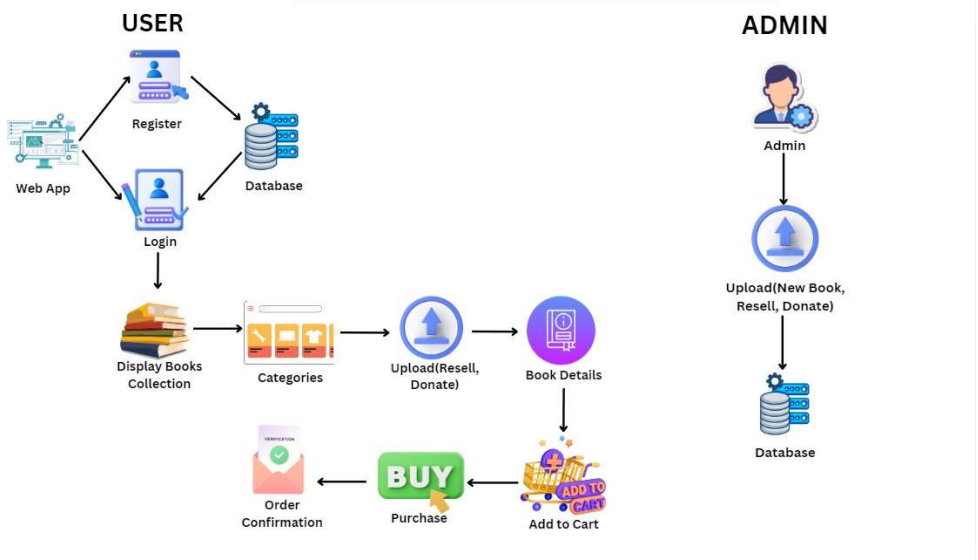
8. Shipping Logistics:

Problem: How can we manage shipping logistics effectively for both purchased and donated books?

Solution: Partner with local delivery services to offer flexible, affordable shipping options and provide users with tracking and estimated delivery times.

This system ensures reliable delivery for buyers and allows donations to be easily coordinated by the platform or recipients, creating a smooth process for everyone involved.

3.5. ARCHITECTURE DESIGN:



1. User Authentication: This component verifies user credentials, managing secure logins and registrations. It enables access control by validating user identities to provide a personalized experience across the platform.

2. Web Application: Serving as the platform's core, the web application processes requests, manages sessions, and orchestrates interactions among different components. It ensures seamless communication between the server, database, and client interfaces.

3. Interface of Site: This is the front-facing UI/UX layer, presenting users with an intuitive, responsive design. It enables smooth navigation and provides easy access to all services, such as browsing categories, viewing books, and managing accounts.

4. Categories: Organized into publishers, books, and uploads, this module allows users to explore available books, upload e-notes (earning rewards), or manage activities like book reselling and donation tracking. It personalizes content based on user interests and activity.

5. Display Books: Here, the system displays detailed information about books, enabling users to view covers, descriptions, and pricing. It also allows them to preview content and make informed purchasing decisions.

6. Add to Cart: Users can add selected books to a shopping cart, which temporarily stores their choices while they continue browsing. This component facilitates book quantity adjustments, saving for later, and provides a checkout gateway.

7. Payment: Integrated with secure payment gateways, this section processes transactions through methods like credit cards, e-wallets, or bank transfers. It ensures transaction security with encryption and user-friendly error handling.

8. Email Confirmation: Following a successful purchase, this component sends an automated email receipt confirming the order details, providing peace of mind and proof of transaction for the user.

3.6. DESCRIPTION OF MODULES:

3.6.1. USER AUTHENTICATION

The User Authentication module is critical for maintaining security and privacy across the platform. This module handles user registration, login, password management, and session maintenance. Using industry-standard practices like hashing passwords, two-factor authentication, and CAPTCHA, it ensures that only verified users can access the system, safeguarding user data from unauthorized access.

Beyond basic authentication, this module tracks user roles and permissions, such as general users, administrators, or contributors, to deliver role-based access. For example, an admin may have privileges to manage content, while a regular user can only browse and purchase books. The module's flexible permission settings make it adaptable for expanding roles or privileges as the application grows in functionality.

Finally, the User Authentication module plays a crucial role in

maintaining active sessions. It handles session expiration and prompts users to re-authenticate if the session is inactive for a long period, balancing security with a seamless user experience. This module works in coordination with the web application core to ensure that only authenticated requests reach sensitive areas of the platform, adding an essential security layer to the system.

3.6.2. WEBPAGE

The Interface of Site, or frontend, serves as the visual and interactive layer, facilitating user interaction with the platform. Built with responsive design principles, this module adapts the platform for use across devices, from desktops to mobile phones. The frontend implements an intuitive layout, helping users effortlessly navigate categories, view books, access their cart, and complete transactions.

This module is designed with accessibility in mind, offering a user-friendly experience for all users, including those with disabilities. By employing accessible fonts, colors, and navigation structures, it ensures that all users can easily interact with each function. Additionally, it provides real-time feedback, such as loading indicators and validation messages, enhancing the user experience and building user confidence as they interact with various features.

Dynamic elements on the frontend allow for a personalized experience, from highlighting new or popular books to offering suggestions based on user preferences. It's tightly integrated with the backend for data updates, making interactions like adding to cart or managing account settings seamless and efficient. Through robust frontend design, this module is crucial for creating a consistent and engaging user journey.

3.6.3 CATEGORIES

The Categories module organizes content, enabling users to browse books, publishers, and upload-related options like reselling, donating, and e-note submissions. This module supports easy exploration by grouping similar items, such as publishers or genres, and enhances discoverability, offering filters and

search functions. By enabling users to find content efficiently, it increases engagement and purchase likelihood.

- **Display Books**

  The Display Books module is where users can view detailed information about individual books. This module fetches and displays book metadata, including descriptions, ratings, author information, pricing, and availability. Through high-resolution images, preview options, and user reviews, it allows users to make well-informed choices, adding value to the browsing experience.For book listings, the module displays detailed metadata—author, publication date, genre, price, and more—offering users comprehensive information at a glance.

- **Shopping Cart**

  The Shopping Cart module is a transitional area for users to store items temporarily before proceeding to checkout. It allows users to view selected books, manage quantities, and keep track of total costs. Users can add or remove books, save items for later, or clear the cart, providing flexibility while shopping.

  The cart module also includes dynamic price adjustments, tax calculation, and discount application, so users see the final cost clearly.

- **Upload (E-notes, Resell books, Donators)**

  This module allows for custom content uploads, giving users options to earn rewards through e-note submissions. Each action is processed through user-specific workflows, with clear instructions and tracking capabilities to manage uploads. This setup provides flexibility for users who wish to contribute to the platform as content providers.

3.6.4 EMAIL CONFIRMATION

This email confirmation serves as a critical step to verify users' email addresses, ensuring they are legitimate and that account activity can be securely tied to their email. Including a direct, clickable link makes the confirmation

process convenient, minimizing barriers for users. A simple message of appreciation, such as “Thank you for joining,” establishes a friendly and welcoming tone that enhances user engagement.

After confirming their email, users gain full access to the platform. This includes options to sell books by creating listings, donate books to others, and browse a variety of available books. A verified email helps protect users’ data and enables smoother communication, as confirmations and notifications are sent directly to users’ inboxes. The email also builds excitement and encourages them to take immediate action to finalize their account setup.

Additionally, verifying email addresses adds an extra layer of security, helping to prevent spam accounts and unauthorized access. It’s a best practice in online platforms to avoid fraudulent activity and maintain trust in the user base. By clearly stating what users can do after verification, this email not only confirms their signup but also primes them for meaningful engagement on the platform.

3.7. DATAFLOW DIAGRAM

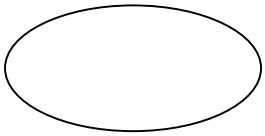
A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system.

These are known as logical data flow diagrams. The physical data flow diagrams show the actual implementation and movement of data between people, departments, and workstations. A full description of a system actually consists of a set of data flow diagrams. The development of DFD is done on several levels.

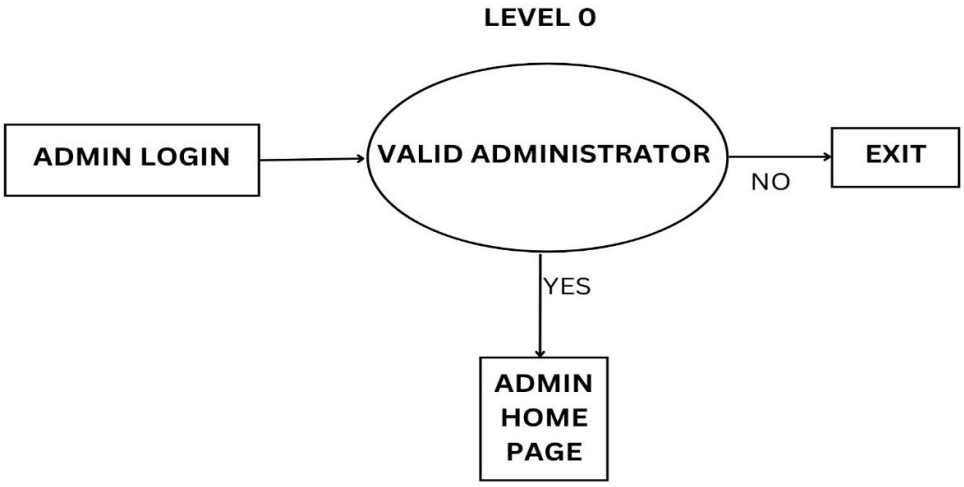
Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for an analyst to understand the process.

Source or Destination of

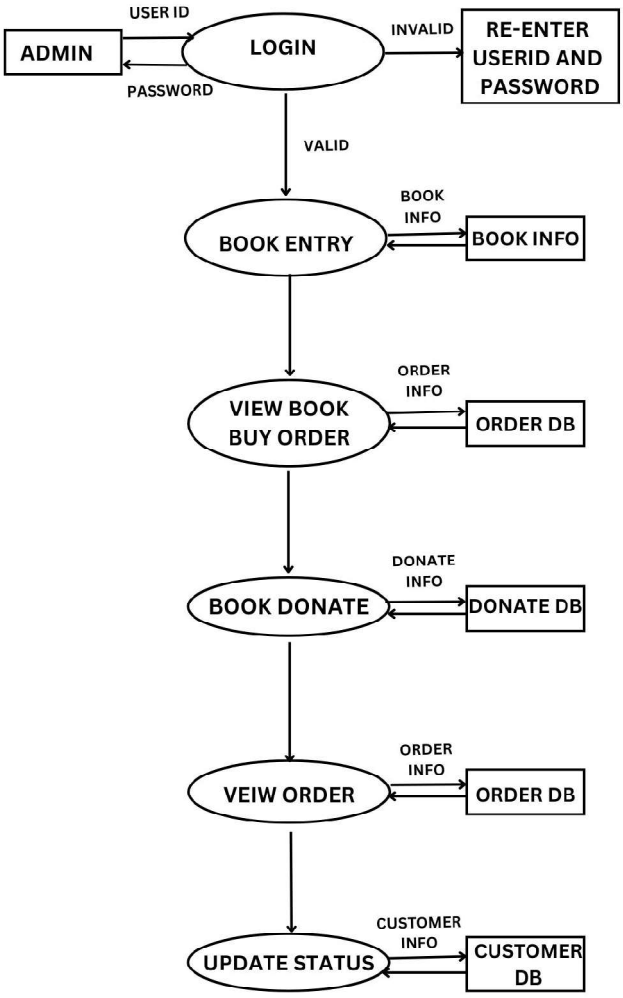
Data Flow



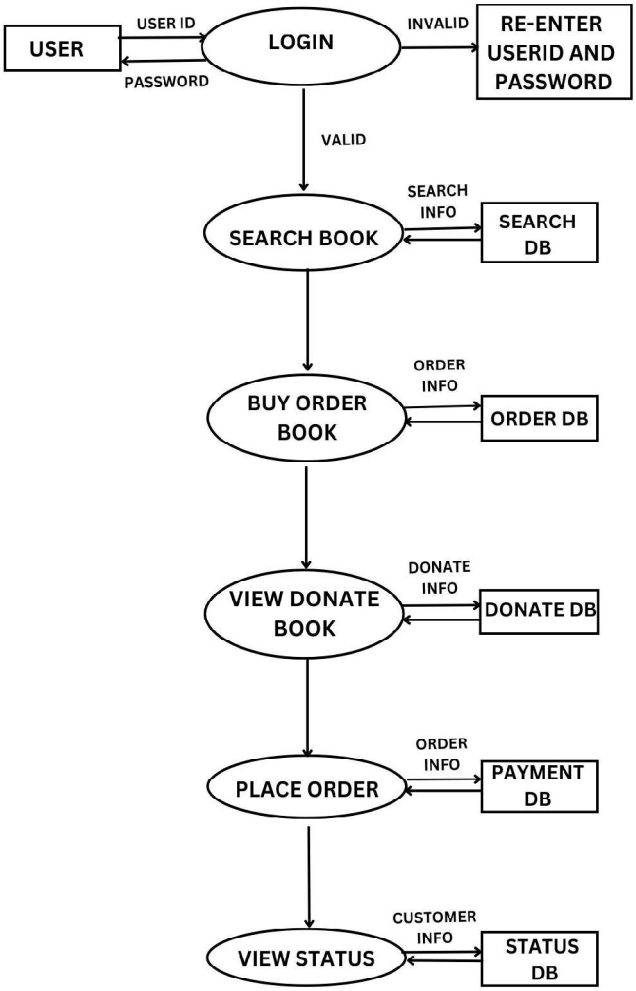
Process



LEVEL 1



LEVEL 2



CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

- Processor: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz
- Installed RAM:8.00 GB (7.84 GB usable)
- System type: 64-bit operating system, x64-based processor

4.2 SOFTWARE REQUIREMENTS:

- Operating System: Windows
- Back End: PHP
- Front End : HTML,CSS,JAVASCRIPT
- IDE: Visual Studio Code
- Web Server: xammp
- Database: MYSQL

4.3 APPLICATION REQUIREMENTS:

HTML: HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within a tag which defines the structure of web pages. The language is used to annotate text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

CSS: Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows applying styles to web pages.

CSS enables to do this independent of the HTML that makes up each web page. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

JAVASCRIPT(JS): JavaScript is a lightweight, cross-platform, interpreted scripting language. It is well-known for the development of web pages, and many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server side developments. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

PHP: The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension ".php".

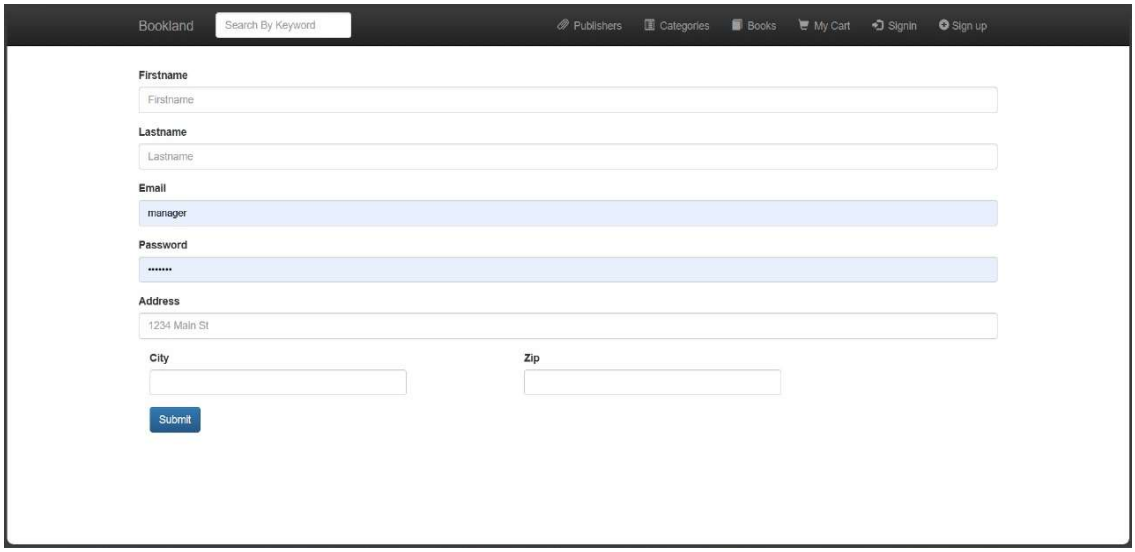
MY SQL: MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.

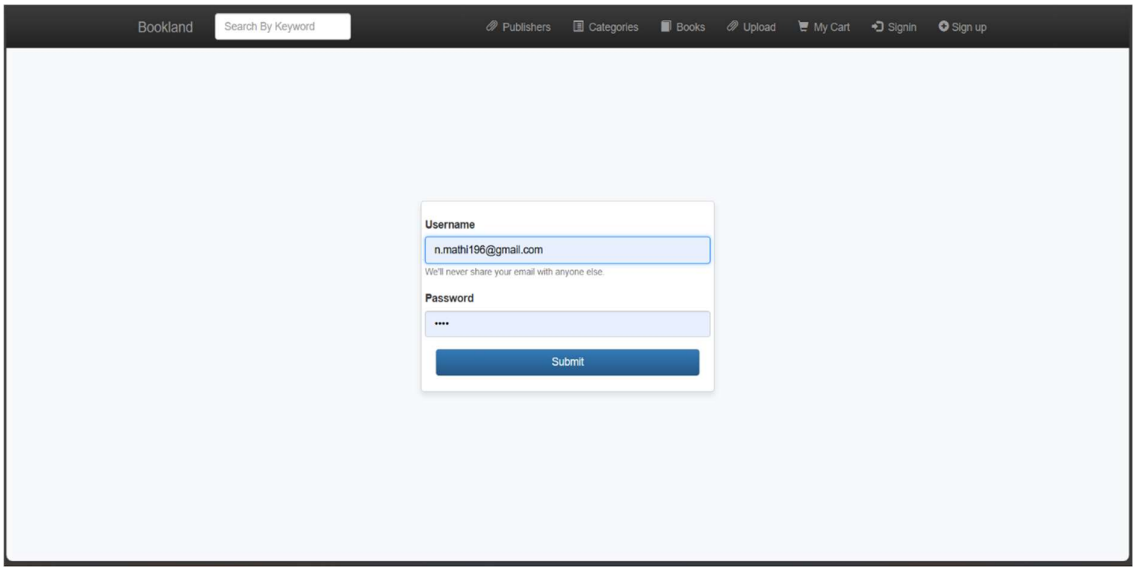
CHAPTER 5

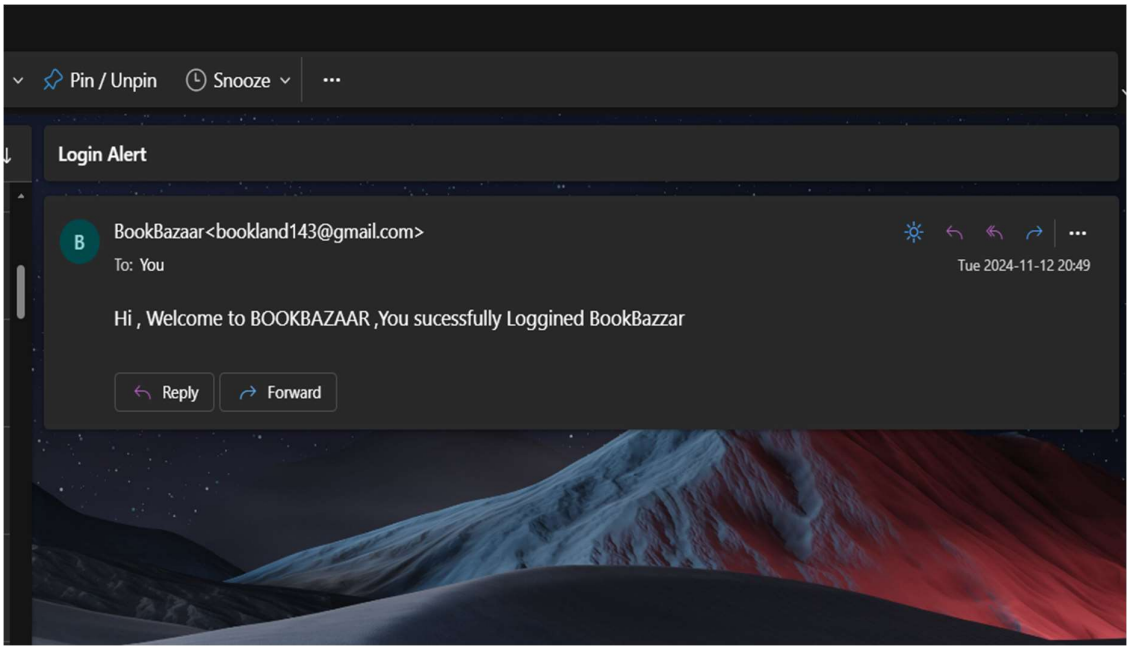
IMPLEMENTATION

5.1 USER AUTHENTICATION

The User Authentication module includes registration, login, password management, and session handling. Registration and login forms are created using HTML, and data submission is managed by PHP scripts, which interact with the MySQL database for user verification. PHP's password_hash() function securely encrypts passwords before storing them in MySQL, where user data like username, password, and role permissions are stored. Secure storage of passwords with hashing. Role-based access control for differentiated user permissions.

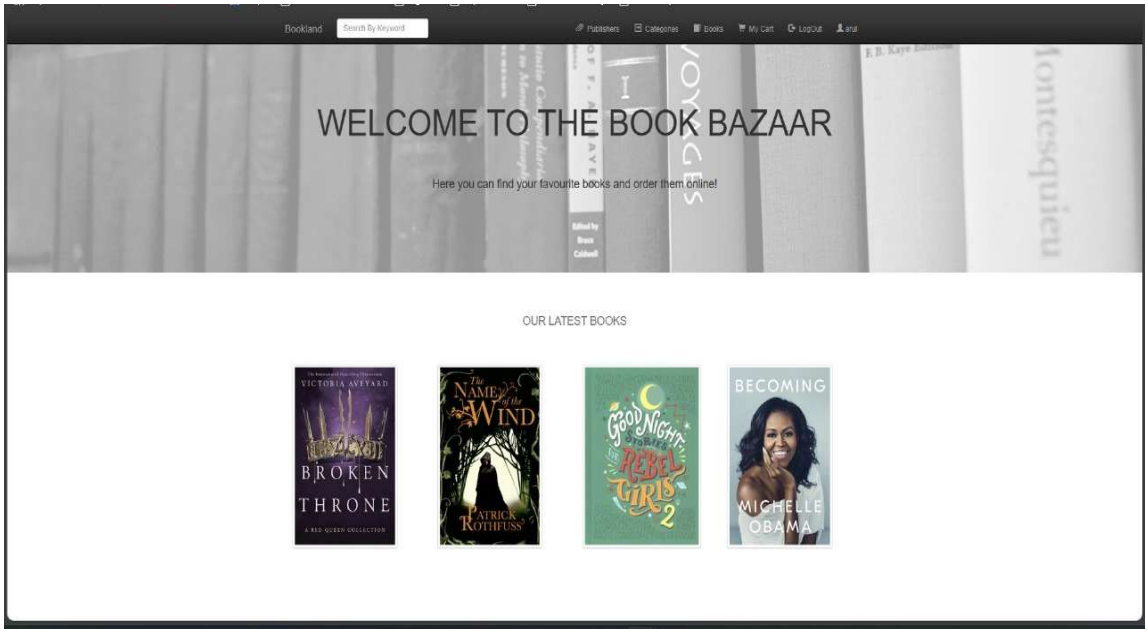






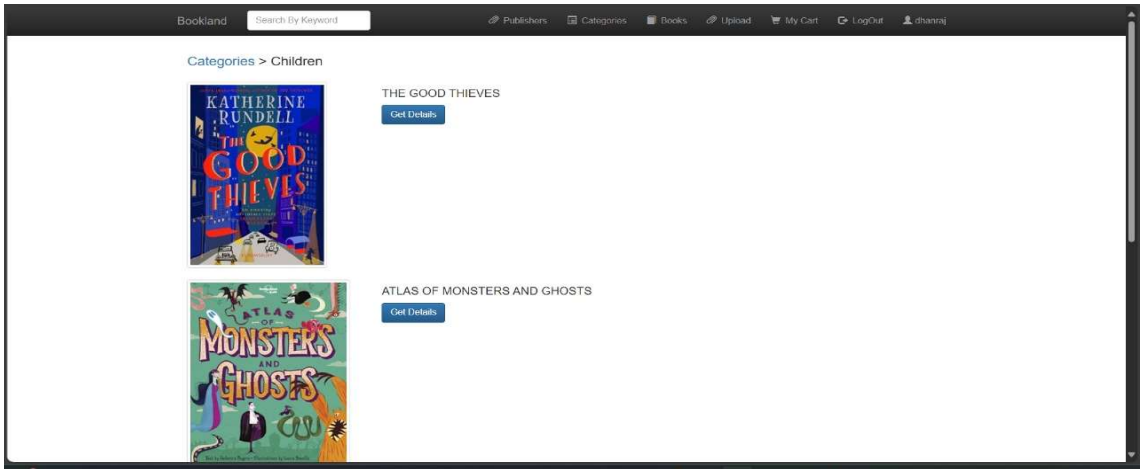
5.2 WEB PAGE

The Web Application Core module, powered by PHP, manages routing, session handling, and overall functionality integration. The index.php file acts as the entry point, while other PHP scripts handle specific actions like viewing products, managing the cart, and handling payments. MySQL databases are accessed with SQL queries within PHP to load content dynamically based on user requests. Central routing through index.php to manage requests. Caching for frequently accessed data to improve performance. Error handling and logging for efficient debugging.



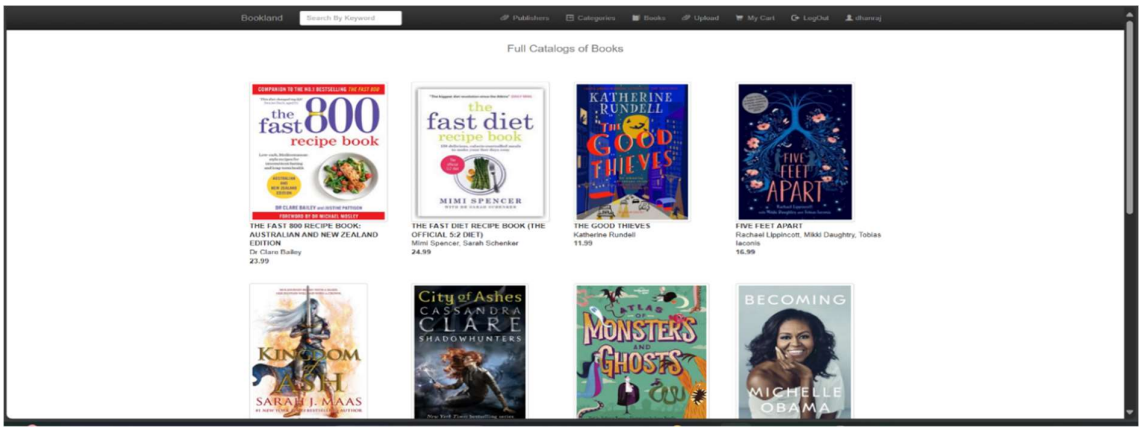
5.3 CATEGORIES

The Categories module, developed using PHP and MySQL, organizes and retrieves content based on user-selected categories (e.g., books by publishers, genres). PHP scripts query MySQL databases, and the results are displayed in HTML. Uploads and user-specific actions (like reselling books) are processed via HTML forms, where PHP handles validation and stores data accordingly. Organized browsing by categories (publishers, genres). Secure upload forms for user submissions. Search and filter options powered by SQL queries.



5.4 DISPLAY BOOKS

The Display Books module fetches and displays detailed book information. PHP scripts interact with MySQL to fetch relevant data, which is formatted in HTML for a user-friendly display. CSS is used to style book listings,, allowing users to dynamically add items to view additional details without refreshing the page. Dynamic book displays with AJAX for interactivity.

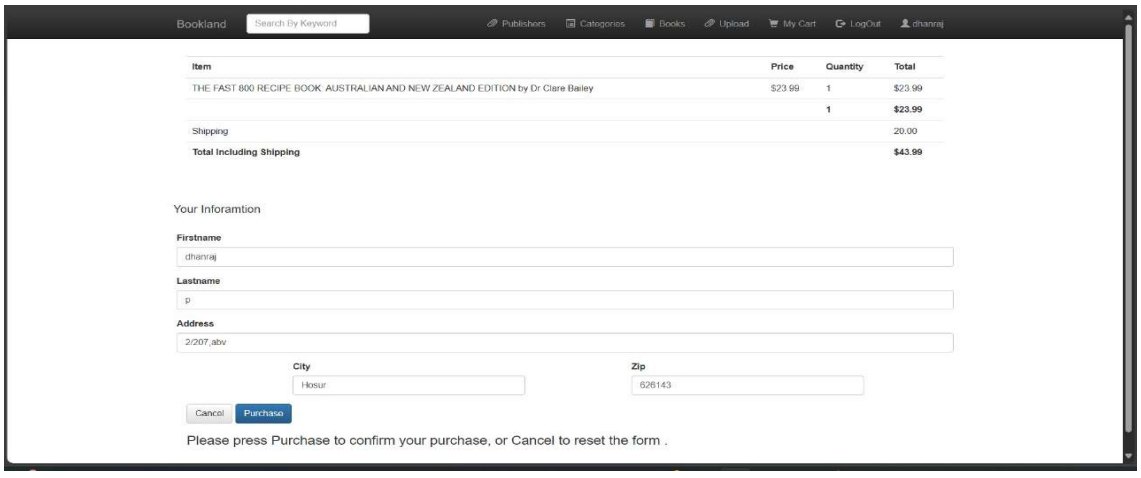


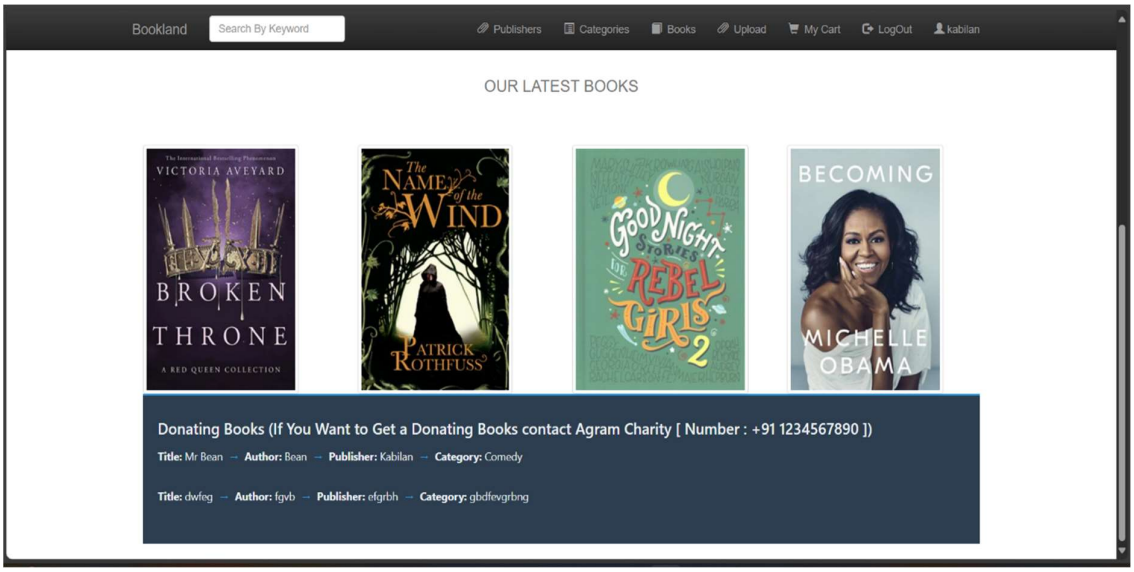
5.5 SHOPPING CART

The Shopping Cart module temporarily stores user selections in $_SESSION variables until checkout. PHP scripts manage adding, removing, and updating cart items. HTML and CSS provide structure and styling, while JavaScript can be used for real-time cart updates, making the experience seamless. MySQL stores cart information for logged-in users to ensure data persistence. Session-based cart management for temporary data storage.

Dynamic cart total calculation and item management. Smooth transition to checkout with persistent data for logged-in users. The Shopping Cart module integrates session-based storage for temporary cart management, database storage for logged-in user persistence, and real-time updates using JavaScript for a seamless experience. This approach ensures efficient and responsive cart handling, allowing users to interact with their selections easily and transition smoothly to checkout with consistent and reliable data.

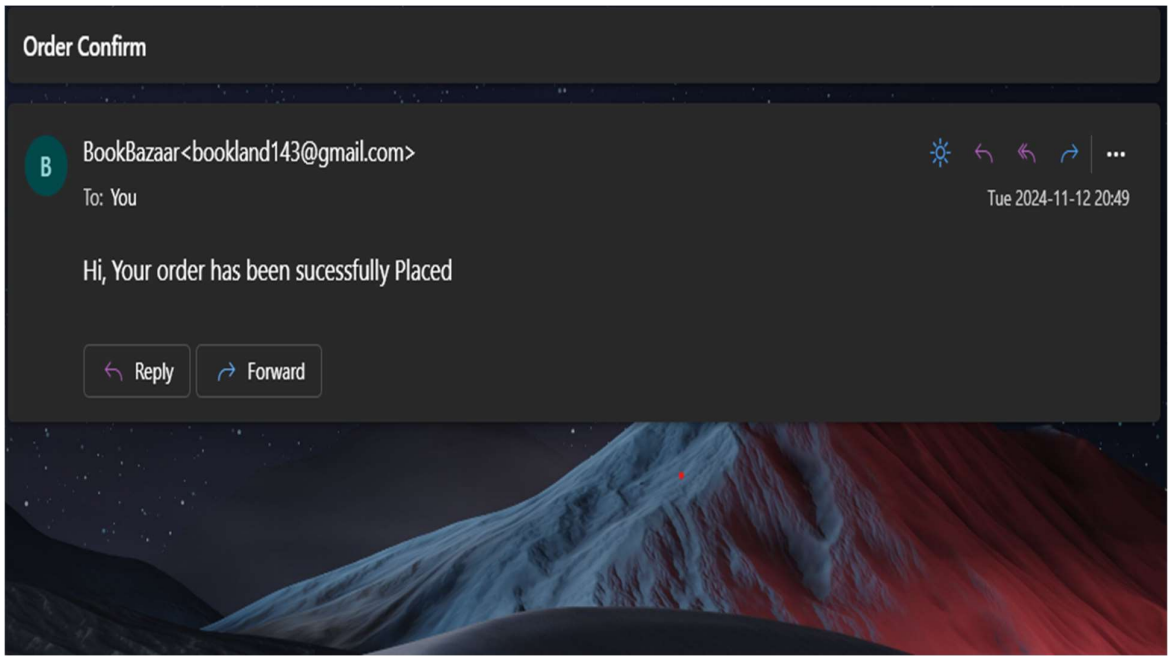
The user begins by logging into their account. Once logged in, the donor navigates to the "Donate" section of the platform. This dedicated feature allows users to easily access donation-related options and start the process of listing their books for donation. In this step, the user provides essential details about the book they wish to donate. Information includes: Title, Author, and Genre: Basic information for categorizing and identifying the book. The user can upload images of the book to give potential recipients a clear view of its condition.





5.6. EMAIL CONFIRMATION

This module uses PHP's mail function or external SMTP libraries (like PHPMailer) to send confirmation emails after a successful order. Emails contain order details and are stored in MySQL for historical tracking. HTML templates style the email, while PHP dynamically fills in content like order items and totals. Automated email confirmation with dynamic details. Email templates for consistent, branded communication. Integration with MySQL for tracking email history.



5.7. SOFTWARE DESCRIPTION

1. VISUAL STUDIO

Visual Studio Code (VS Code) is a lightweight, open-source Integrated Development Environment (IDE) developed by Microsoft. It is highly versatile and customizable, making it ideal for a wide range of development projects, including full-stack applications. Here's an overview of Visual Studio Code: User Interface: VS Code provides an intuitive and highly customizable interface. It supports multiple panels for editing, debugging, and source control, making project management seamless. Code Editor: The code editor in VS Code offers features like IntelliSense, syntax highlighting, and real-time error detection. It supports numerous programming languages such as JavaScript, TypeScript, HTML, and CSS. Extensions: The VS Code Marketplace offers thousands of extensions that enhance functionality, including tools for linting, debugging, and integration with various frameworks and databases like MongoDB.Version Control: Visual Studio Code has built-in Git integration, allowing developers to manage repositories, commit changes, and push code directly from the IDE. Debugging: VS Code includes a built-in debugger that supports Node.js, making it easier to debug backend code and solve runtime issues efficiently.

Cross-Platform Support: As a cross-platform tool, VS Code works on Windows, macOS, and Linux, making it a versatile choice for developers on any operating system. Visual Studio Code is ideal for full-stack development, providing a balance between performance and a wide range of features for developers.

2. XAMPP

XAMPP (Cross-Platform Apache, MariaDB, PHP, and Perl) is an open-source web server solution stack package, developed by Apache Friends. It is a cross-platform tool that provides an easy-to-install package with Apache server, MariaDB (database), PHP, and Perl, designed to help developers run a local server environment for building, testing, and deploying web applications. For

"Book Bazaar: Online Book Reselling and Donating," XAMPP provides a local development environment that enables developers to simulate a web server on their personal computer. XAMPP's inclusion of Apache and MySQL (through MariaDB) makes it a perfect tool for setting up, developing, and testing PHP-based applications like Book Bazaar.

Apache Server: Acts as the web server, handling HTTP requests and delivering website content to users. MariaDB: The MySQL-compatible database system stores essential data such
as user accounts, book listings, transaction records, and more. As a widely used open-source web server, it handles incoming HTTP requests from users' browsers, interpreting them, and serving the requested web pages, images, scripts, and other assets efficiently. Apache's versatility allows it to handle various types of content requests, from static files to dynamic web pages generated by server-side scripts. It can also manage multiple domains and handle a high volume of concurrent requests, making it a reliable choice for many web applications. Configurable with modules and extensions, Apache can be customized to enhance security, load balancing, and user authentication, ensuring that web services run smoothly and efficiently.

PHP: The scripting language used to write server-side code, such as handling form submissions, user authentication, and other backend functionality. Cross-Platform Compatibility: XAMPP runs on Windows, macOS, and Linux, providing flexibility for developers working across different operating systems. Ease of Installation and Use: XAMPP simplifies the setup of a complete server stack, which includes Apache, MariaDB, and PHP, allowing developers to run their projects locally without complex configurations.

phpMyAdmin Integration: XAMPP includes phpMyAdmin, a web-based interface for managing MariaDB databases. It simplifies database operations, like creating tables, inserting data, and running queries. Development Environment: XAMPP allows developers to simulate a real server environment, ideal for testing the Book Bazaar website before deployment.

Control Panel: The XAMPP Control Panel makes it easy to start and stop individual services like Apache and MariaDB, as well as monitor their status. Local Testing: XAMPP allows developers to test all project features such as user authentication, book listings, donations, and transactions—locally, ensuring that everything works correctly before deployment.

Database Management: Through phpMyAdmin, developers can easily manage the Book Bazaar database structure, user data, book records, and transactions without needing advanced command-line skills. Error Debugging: XAMPP's local server environment allows developers to test error handling, troubleshoot issues, and refine code in real-time.

Operating System: Compatible with Windows, macOS, and Linux.Storage: Requires around 600 MB for installation, but additional space may be needed for databases and other project files. Memory: A minimum of 1 GB RAM is recommended for optimal performance. Password Protection: XAMPP allows enabling password protection for phpMyAdmin, which is essential during development.

5.8. CODE IMPLEMENTATION

Step 1: Set up the registration page

The User Registration in our application is a critical component that facilitates the onboarding of students and teachers, allowing them to create accounts and access the platform's educational resources. This module is designed to be user-friendly and secure, ensuring a smooth registration process. Registration and login forms are created using HTML, and data submission is managed by PHP scripts, which interact with the MySQL database for user verification. Here's the code to setup the registration page:

Code Snippet:

```
<?php
   $title = "User SignUp";
   require_once "./template/header.php";
?>

   <form class="form-horizontal" method="post" action="user_signup.php">
   <div class="form-group">
      <label for="exampleInputEmail1">Firstname</label>
           <input type="text" class="form-control" aria-describedby="emailHelp" placeholder="Firstname"
name="firstname">
   </div>
   <div class="form-group">
      <label for="exampleInputEmail1">Lastname</label>
           <input type="text" class="form-control" aria-describedby="emailHelp" placeholder="Lastname"
name="lastname">
   </div>
   <div class="form-group">

      <label for="inputEmail4">Email</label>
      <input type="text" class="form-control" id="inputEmail4" placeholder="Email" name="email">
      </div>
      <div class="form-group">
      <label for="inputPassword4">Password</label>
            <input type="password" class="form-control" id="inputPassword4" placeholder="Password"
name="password">
   </div>
   <div class="form-group">
      <label for="inputAddress">Address</label>
      <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St" name="address">
   </div>
   <div class="form-row">
      <div class="form-group col-md-4">
      <label for="inputCity">City</label>
      <input type="text" class="form-control" id="inputCity" name="city">
      </div>
      <div class="form-group col-md-2">
      </div>
      <div class="form-group col-md-4">
      <label for="inputZip">Zip</label>
      <input type="text" class="form-control" id="inputZip" name="zipcode">
      </div>
   </div>
   <div class="form-group col-md-12">
   <button type="submit" class="btn btn-primary">Submit</button>
   </div>
</form>
<div style="position:fixed; bottom:120px">

<?php
   $fullurl="http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
   if(strpos($fullurl,"signup=empty")==true)
{
      echo '<P style="color:red">You did not fill in all the fields.</P>';
      exit();
   }
?>
</div>
<?php require_once "./template/footer.php";>?
```

Step 2: Set up the home page

The home page is structured to dynamically pull and display content based on user interactions and available database records. For instance, the "Featured Books" section on the home page pulls a subset of book listings from the MySQL database and displays each with a title, author, brief description, and condition. Additionally, users who are logged in will see personalized elements on the home page, such as recommended books based on their browsing and purchasing history, created by querying data specific to each user. Overall, the code implementation of the home page focuses on delivering a seamless, engaging experience to attract and retain users.

Code Snippet:

```
<?php
 session_start();
 $count = 0;
 // connecto database
 $title = "Index";
 require_once "./template/header.php";
 require_once "./functions/database_functions.php";
 $conn = db_connect();
 $row = select4LatestBook($conn);
?>
    <br/> <br/>
     <p class="lead text-center text-muted">OUR LATEST BOOKS</p>
     <br><br>
     <div class="row">
       <?php foreach($row as $book) { ?>
       <div class="col-md-3">
         <a href="book.php?bookisbn=<?php echo $book['book_isbn']; ?>">
            <img class="img-responsive img-thumbnail" src="./bootstrap/img/<?php echo $book['book_image'];
?>">
         </a>
       </div>
       <?php } ?>
     </div>
<?php
 if(isset($conn)) {
  mysqli_close($conn);} require_once "./template/footer.php";?>
```

Step 3: set up the display books

Pagination or lazy loading may be implemented to handle large numbers of books efficiently, ensuring fast load times and a smooth browsing experience.

Code Snippet:

```
<?php
 session_start();
 $count = 0;
 // connecto database
 require_once "./functions/database_functions.php";
 $conn = db_connect();
 if(isset($_POST['title'])){
   if(isset($_POST['asc'])){
     $query = "SELECT * FROM books order by book_title asc";

   }
   else if(isset($_POST['desc'])){
     $query = "SELECT * FROM books order by book_title desc";
   }else{
     $query = "SELECT * FROM books";
   }
 }else if(isset($_POST['price'])){
   if(isset($_POST['asc'])){
     $query = "SELECT * FROM books order by book_price asc";

   }
   else if(isset($_POST['desc'])){
     $query = "SELECT * FROM books order by book_price desc";
   }else{
     $query = "SELECT * FROM books";
   }
 }else if(isset($_POST['author'])){
   if(isset($_POST['asc'])){
     $query = "SELECT * FROM books order by book_author asc";

   }
   else if(isset($_POST['desc'])){
     $query = "SELECT * FROM books order by book_author desc";
   }else{
     $query = "SELECT * FROM books";
   }
 }else{
   $query = "SELECT * FROM books";
 }

 $result = mysqli_query($conn, $query);
 $title = "Full Catalogs of Books";
   require_once "./template/header.php";
?>

 <p class="lead text-center text-muted">Full Catalogs of Books</p>

<br><br>

   <?php for($i = 0; $i < mysqli_num_rows($result); $i++){ ?>
     <div class="row">
       <?php while($query_row = mysqli_fetch_assoc($result)){ ?>
         <div class="col-md-3">
           <a href="book.php?bookisbn=<?php echo $query_row['book_isbn']; ?>">
                                <img class="img-responsive img-thumbnail" src="./bootstrap/img/<?php echo
$query_row['book_image']; ?>">
```

```
            </a>
            <table>
              <tr>
                <td><strong>  <?php echo $query_row['book_title']; ?></strong></td>
              </tr>
              <tr>
              <td> <?php echo $query_row['book_author']; ?> <br>
              <?php echo $query_row['type']; ?>
            </td>
              </tr>
              <tr>
              <td><strong><?php echo $query_row['book_price'];?></strong>  </td>
              </tr>
            </table>
          </div>
        <?php
          $count++;
          if($count >= 4){
             $count = 0;
             break;
           }
          } ?>
      </div>
      <br><br>
<?php
      }
  if(isset($conn)) { mysqli_close($conn); }
  require_once "./template/footer.php";
?>
 <script  src=https://code.jquery.com/jquery-3.2.1.slim.min.js integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"crossorigin="ano
nymous"></script>
 <script  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script   src=https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js          integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
```

Step 4: Set up the Shopping cart page

The shopping cart provides an interactive and responsive experience, allowing users to add, view, update, and remove books before proceeding to checkout. When a user selects a book to buy, the item is added to the cart using JavaScript to handle immediate updates without reloading the page, improving the user experience.

Each book entry in the cart displays its title, price, quantity selection, and subtotal, which updates dynamically as quantities are adjusted. The database is utilized to store cart data, ensuring the cart persists even if the user navigates.

Code Snippet:

```
<?php
    session_start();
    require_once "./functions/database_functions.php";
    require_once "./functions/cart_functions.php";
    $conn = db_connect();
    // book_isbn got from form post method, change this place later.
    if(isset($_POST['bookisbn'])){
        $book_isbn = $_POST['bookisbn'];
    }
    if(isset($book_isbn)){
        // new iem selected
        if(!isset($_SESSION['cart'])){
            // $_SESSION['cart'] is associative array that bookisbn => qty
            $_SESSION['cart'] = array();
            $_SESSION['total_items'] = 0;
            $_SESSION['total_price'] = '0.00';
        }
        if(!isset($_SESSION['cart'][$book_isbn])){
            $_SESSION['cart'][$book_isbn] = 1;
        } elseif(isset($_POST['cart'])){
            $_SESSION['cart'][$book_isbn]++;
            unset($_POST);
        }
    }
    // if save change button is clicked , change the qty of each bookisbn
    if(isset($_POST['save_change'])){
        foreach($_SESSION['cart'] as $isbn =>$qty){
            if($_POST[$isbn] == '0'){
                unset($_SESSION['cart']["$isbn"]);
            } else {
                $_SESSION['cart']["$isbn"] = $_POST["$isbn"];
            }
        }
    }
    // print out header here
    $title = "Your shopping cart";
    require "./template/header.php";
    if(isset($_SESSION['cart']) && (array_count_values($_SESSION['cart']))){
        $_SESSION['total_price'] = total_price($_SESSION['cart']);
        $_SESSION['total_items'] = total_items($_SESSION['cart']);
?>
    <form action="cart.php" method="post">
        <table class="table">
            <tr>
                <th>Item</th>
                <th>Price</th>
                <th>Quantity</th>
                <th>Total</th>
            </tr>
            <?php
                foreach($_SESSION['cart'] as $isbn => $qty){
                    $conn = db_connect();
                    $book = mysqli_fetch_assoc(getBookByIsbn($conn, $isbn));
            ?>
            <tr>
                <td><?php echo $book['book_title'] . " by " . $book['book_author']; ?></td>
                <td><?php echo  $book['book_price']; ?></td>
                <td><input type="text" value="<?php echo $qty; ?>" size="2" name="<?php echo $isbn; ?>"></td>
                <td><?php echo  $qty * $book['book_price']; ?></td>
            </tr>
            <?php } ?>
            <tr>
```

```
            <th> </th>
            <th> </th>
            <th><?php echo $_SESSION['total_items']; ?></th>
            <th><?php echo  $_SESSION['total_price']; ?></th>
          </tr>
        </table>
             <button type="submit" class="btn btn-primary" name="save_change"><span class="glyphicon glyphicon-ok"></span> Save Changes</button>
    </form>
    <br/><br/>
    <a href="checkout.php" class="btn btn-primary">Go To Checkout</a>
    <a href="books.php" class="btn btn-primary">Continue Shopping</a>
<?php
    } else {
       echo "<p class=\"text-warning\">Your cart is empty! Please make sure you add some books in it!</p>";
    }
    if(isset($_SESSION['user'])){
    $customer=getCustomerIdbyEmail($_SESSION['email']);
    $customerid=$customer['id'];
    $query="SELECT * FROM cart join cartitems join books join customers
            on customers.id='$customerid' and cart.customerid='$customerid' and cart.id=cartitems.cartid and cartitems.productid=books.book_isbn";
     $result=mysqli_query($conn,$query);
     $query2="SELECT * FROM cart Where customerid='$customerid'";
     $result2=mysqli_query($conn,$query2);
     if(mysqli_num_rows($result)!=0){
     echo ' <br><br><br><h4>Your Purchase History</h4><table class="table">
     <tr>
        <th>Item</th>
        <th>Quantity</th>
       <th>Date</th>
     </tr>';
       for($i = 0; $i < mysqli_num_rows($result); $i++){
          while($query_row = mysqli_fetch_assoc($result) AND $q = mysqli_fetch_assoc($result2) ){
             echo '<tr>
             <td>
             <a href="book.php?bookisbn=';
             echo $query_row['book_isbn'];
             echo '">';
                          echo '<img style="height:100px;width:80px"class="img-responsive img-thumbnail" src="./bootstrap/img/';
             echo $query_row['book_image'];
             echo '">';
             echo ' </a>
             </td>
        echo '</table>';
    }
}
?>
<?php
    if(isset($conn)){ mysqli_close($conn); }
    // require_once "./template/footer.php";?>
```

Step 5: Set up the purchase page

The code implementation of the Purchase page in the "Online Book Reselling and Donating" platform involves creating a dynamic PHP-based interface where users can view detailed information about available books and initiate the purchasing process.

Code Snippet:

```
<?php
   session_start();
      require_once "./functions/database_functions.php";
   // print out header here
   $title = "Purchase";
   require "./template/header.php";
   // connect database
   if(isset($_SESSION['cart']) && (array_count_values($_SESSION['cart'])))
{
      $customer = getCustomerIdbyEmail($_SESSION['email']);
   ?>
   <table class="table">
      <tr>
         <th>Item</th>
         <th>Price</th>
         <th>Quantity</th>
         <th>Total</th>
      </tr>
         <?php
            foreach($_SESSION['cart'] as $isbn => $qty){
               $conn = db_connect();
               $book = mysqli_fetch_assoc(getBookByIsbn($conn, $isbn));
         ?>
      <tr>
         <td><?php echo $book['book_title'] . " by " . $book['book_author']; ?></td>
         <td><?php echo "$" . $book['book_price']; ?></td>
         <td><?php echo $qty; ?></td>
         <td><?php echo "$" . $qty * $book['book_price']; ?></td>
      </tr>
      <?php } ?>
      <tr>
         <th> </th>
         <th> </th>
         <th><?php echo $_SESSION['total_items']; ?></th>
         <th><?php echo "$" . $_SESSION['total_price']; ?></th>
      </tr>
      <tr>
         <td>Shipping</td>
         <td> </td>
         <td> </td>
         <td>20.00</td>
      </tr>
      <tr>
         <th>Total Including Shipping</th>
         <th> </th>
         <th> </th>
         <th><?php echo "$" . ($_SESSION['total_price'] + 20); ?></th>
      </tr>
   </table>
   <br>
   <br>
   <h4 style="margin-left:-20px">Your Inforamtion</h4>
   <br>
   <form method="post" action="process.php" class="form-horizontal">
   <div class="form-group">
      <label for="exampleInputEmail1">Firstname</label>
            <input type="text" class="form-control" aria-describedby="emailHelp" value="<?php echo
$customer['firstname']?>" name="firstname">
   </div>
   <div class="form-group">
      <label for="exampleInputEmail1">Lastname</label>
            <input type="text" class="form-control" aria-describedby="emailHelp" value="<?php echo
```

```
$customer['lastname']?>" name="lastname">
   </div>

   <div class="form-group">
      <label for="inputAddress">Address</label>
       <input type="text" class="form-control" id="inputAddress" value="<?php echo $customer['address']?>"
name="address">
   </div>
   <div class="form-row">
   <div class="form-group col-md-2">
      </div>
      <div class="form-group col-md-4">
      <label for="inputCity">City</label>
            <input type="text" class="form-control" id="inputCity" name="city" value="<?php echo
$customer['city']?>">
      </div>
      <div class="form-group col-md-2">
      </div>
      <div class="form-group col-md-4">
      <label for="inputZip">Zip</label>
           <input type="text" class="form-control" id="inputZip" name="zipcode" value="<?php echo
$customer['zipcode']?>">
      </div>
   </div>
   <br>
   <div class="form-group col-md-12" >
      <div class="form-group" >
         <div class="col-lg-10 col-lg-offset-2" style="margin-left:0px">
            <button type="reset" class="btn btn-default">Cancel</button>
            <button type="submit" class="btn btn-primary">Purchase</button>
         </div>
      </div>
   </form>
   <p class="lead">Please press Purchase to confirm your purchase, or Cancel  to reset the form .</p>
<?php
   }
 else {
      echo "<p class=\"text-warning\">Your cart is empty! Please make sure you add some books in it!</p>";
   }
   if(isset($conn))
{
 mysqli_close($conn);
 }
   require_once "./template/footer.php";
?>
```

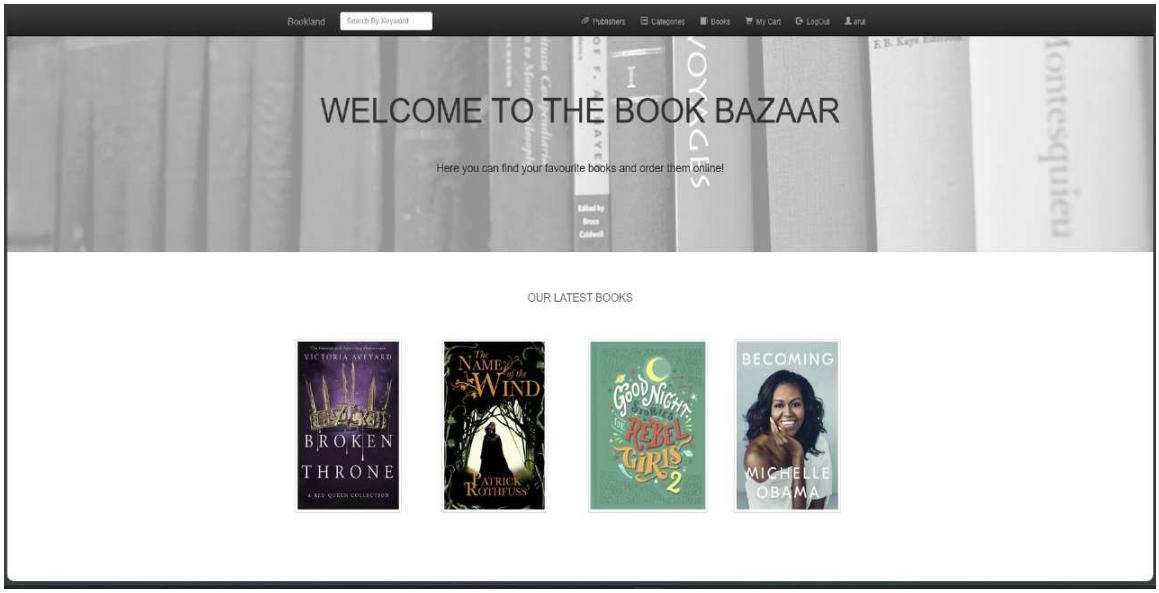
Step 6: Set up the Email confirmation page

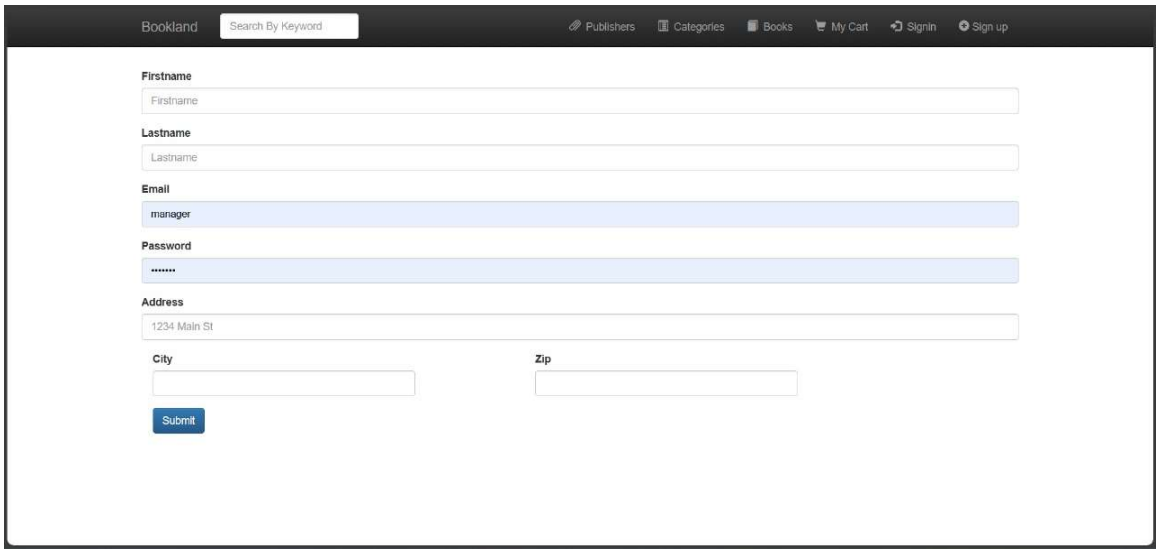
The email confirmation page is implemented to verify the authenticity of a user's email address during the registration or donation process. This feature is crucial to ensure that users have provided a valid email address, facilitating reliable communication for transaction updates and interactions.

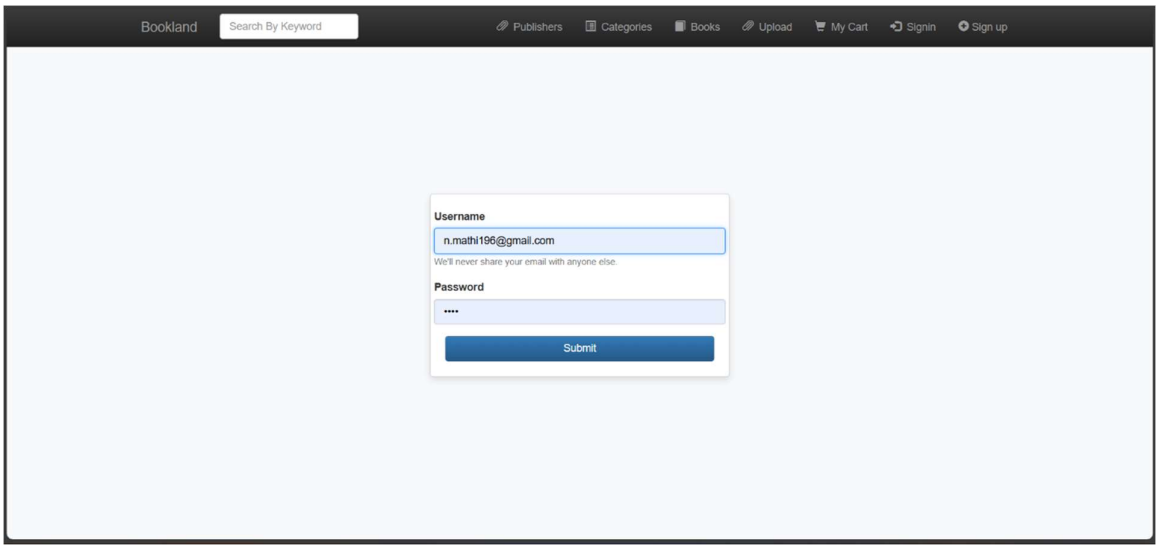
Code Snippet:

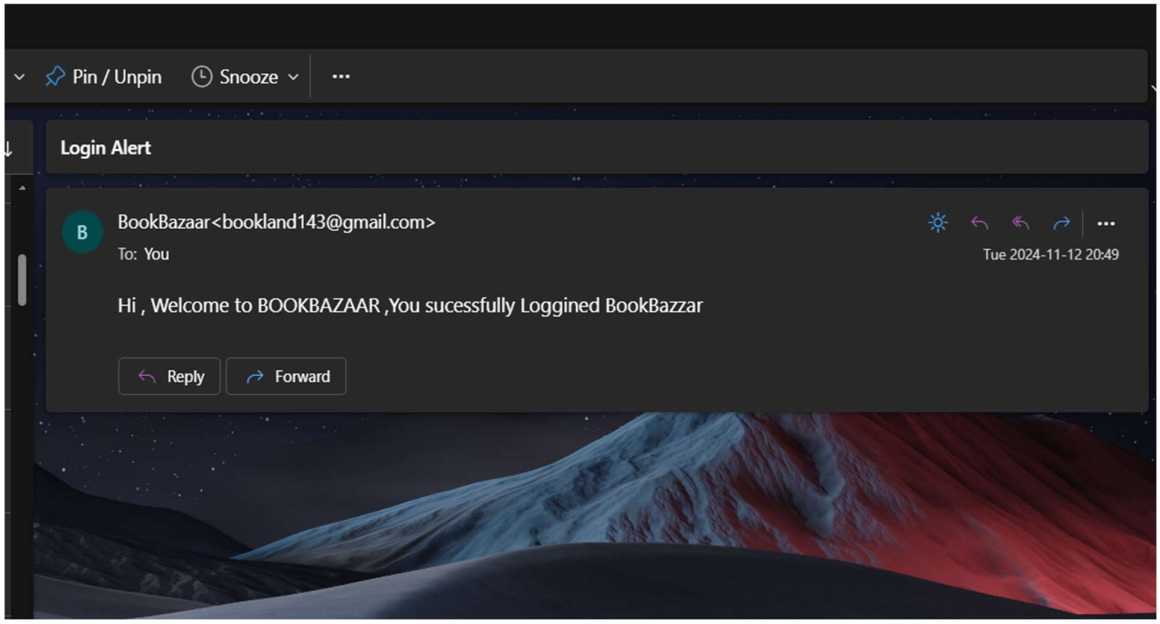
```
<?php
require_once('SMTP.php');
require_once('PHPMailer.php');
require_once('Exception.php');
use \PHPMailer\PHPMailer\PHPMailer;
use \PHPMailer\PHPMailer\Exception;
$mail=new PHPMailer(true);
// Passing `true` enables exceptions
try {
   //settings
   $mail->SMTPDebug=2; // Enable verbose debug output
   $mail->isSMTP(); // Set mailer to use SMTP
   $mail->Host='smtp.gmail.com';
   $mail->SMTPAuth=true; // Enable SMTP authentication
   $mail->Username='donatingnest@gmail.com'; // SMTP username
   $mail->Password='swa2510@'; // SMTP password
   $mail->SMTPSecure='ssl';
   $mail->Port=465;
   $mail->setFrom('support@bookland.com', 'bookland');
   //recipient
   $mail->addAddress('swathimathi5@gmail.com','SWATHI');
   // Add a recipient
   //content
   $mail->isHTML(true);
   // Set email format to HTML
   $mail->Subject='Order Confrm';
   $mail->Body='Hi, Your ordr has been sucessfully Placed';
   $mail->AltBody='This is the body in plain text for non-HTML mail clients';
   $mail->send();
   header('location: index.php');
}
catch(Exception $e)
{
   echo 'Message could not be sent.';
   echo 'Mailer Error: '.$mail->ErrorInfo;
}
?>
```

5.9. RESULT


Bookland
Search By Keyword
Publishers
Categories
Books
My Cart
LogOut
WELCOME TO THE BOOK BAZAAR
Here you can find your favourite books and order them online!
OUR LATEST BOOKS
BROKEN THRONE
The Name of the Wind
PATRICK ROTHFUSS
Good Night Stories for Rebel Girls 2
BECOMING
MICHELLE OBAMA

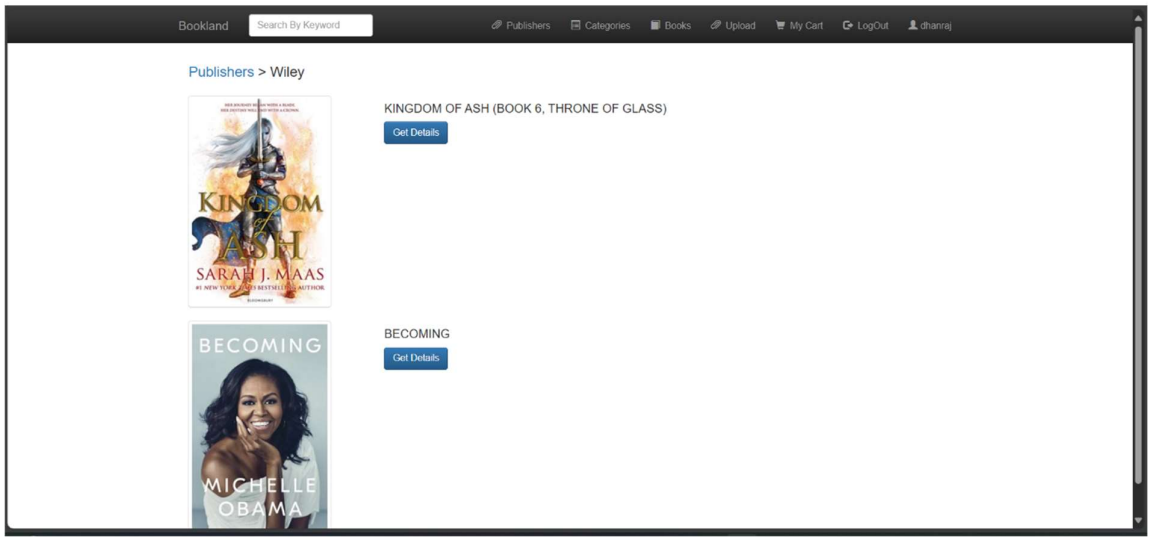

Bookland
Search By Keyword
Publishers
Categories
Books
My Cart
Signin
Sign up
Firstname
Firstname
Lastname
Lastname
Email
manager
Password
Address
1234 Main St
City
Zip
Submit

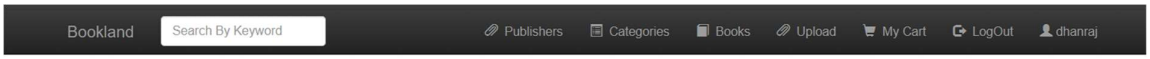

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
Signin
Sign up
Username
n.mathi196@gmail.com
We'll never share your email with anyone else.
Password
Submit



List of Publisher

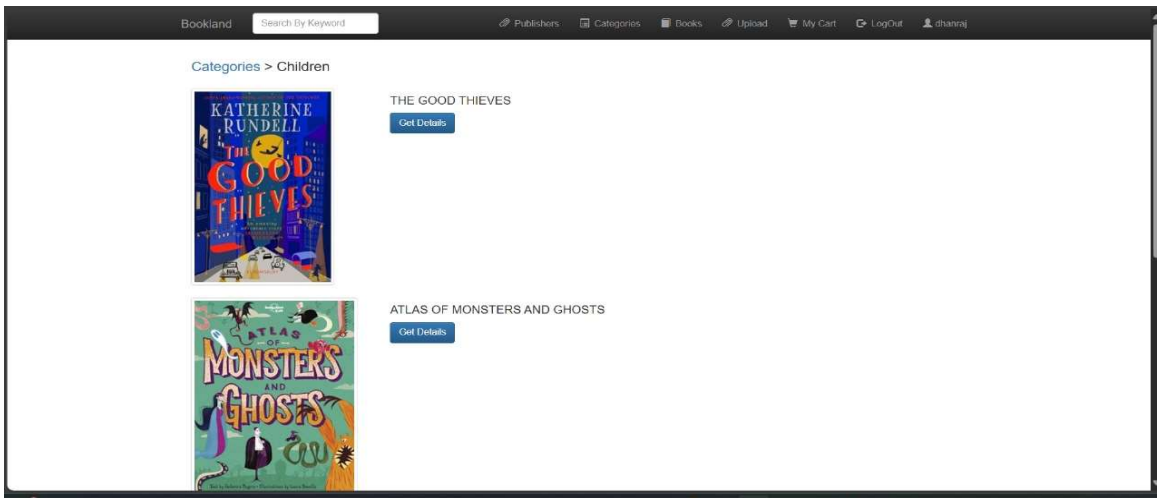
- 2 Wiley
- 4 OReilly Media
- 2 Paper Back
- 1 Bloomsbury Publishing PLC
- 1 Wrox
- List full of books

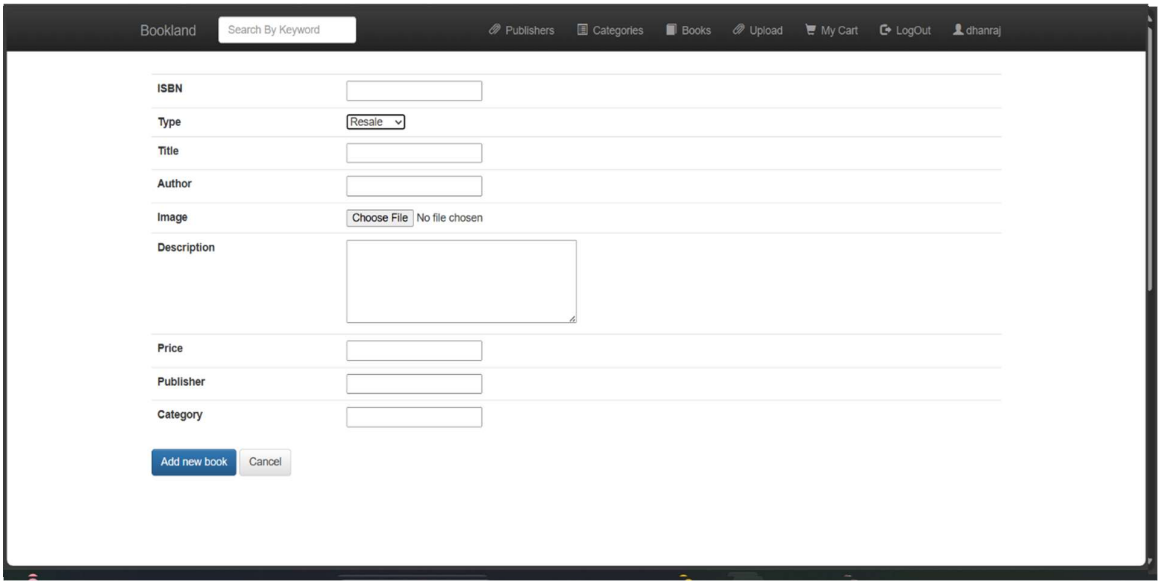


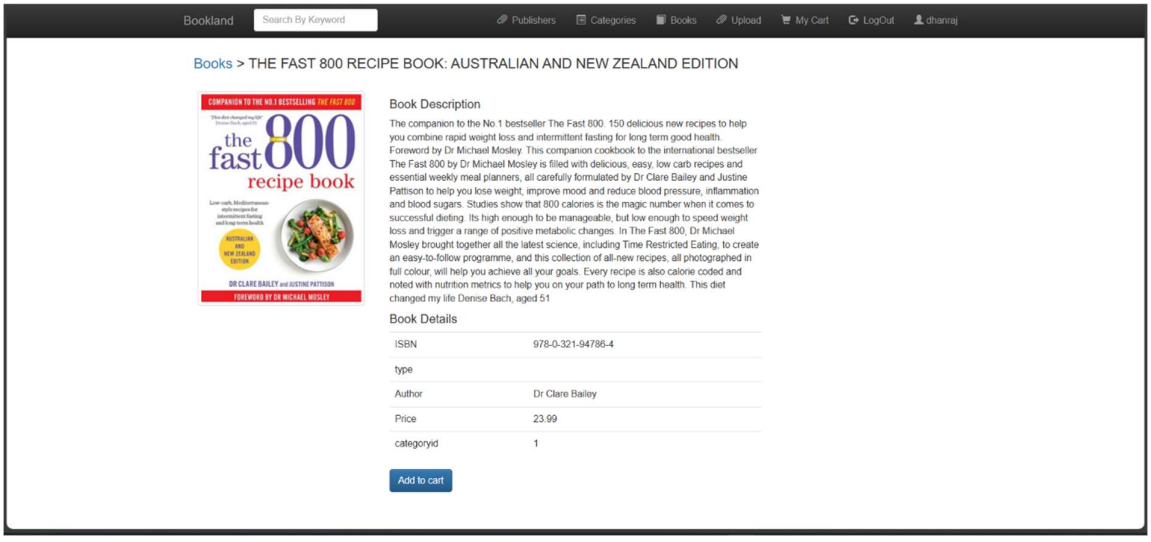


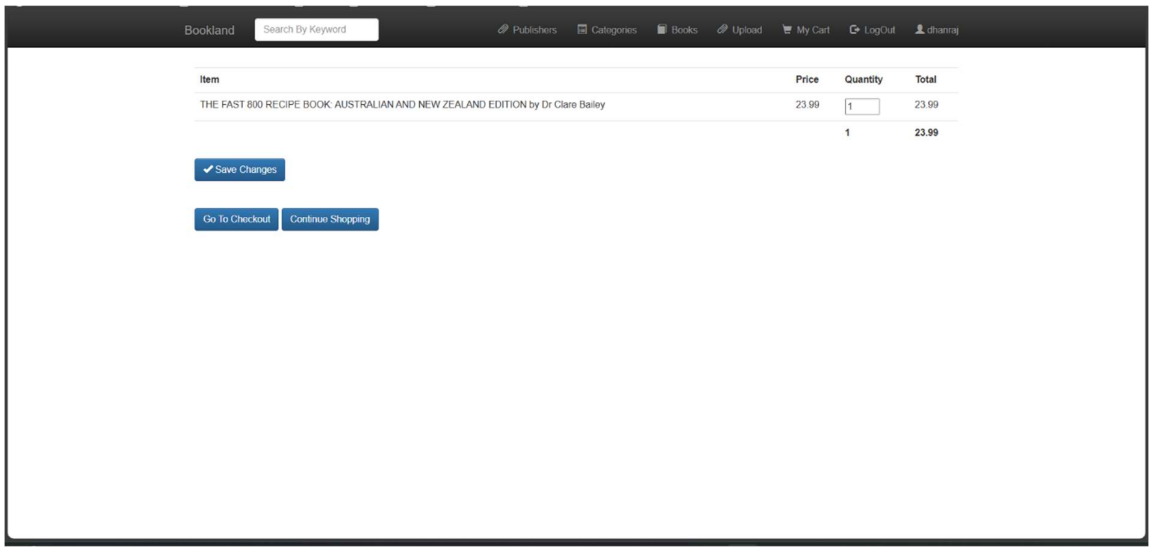
List of Category

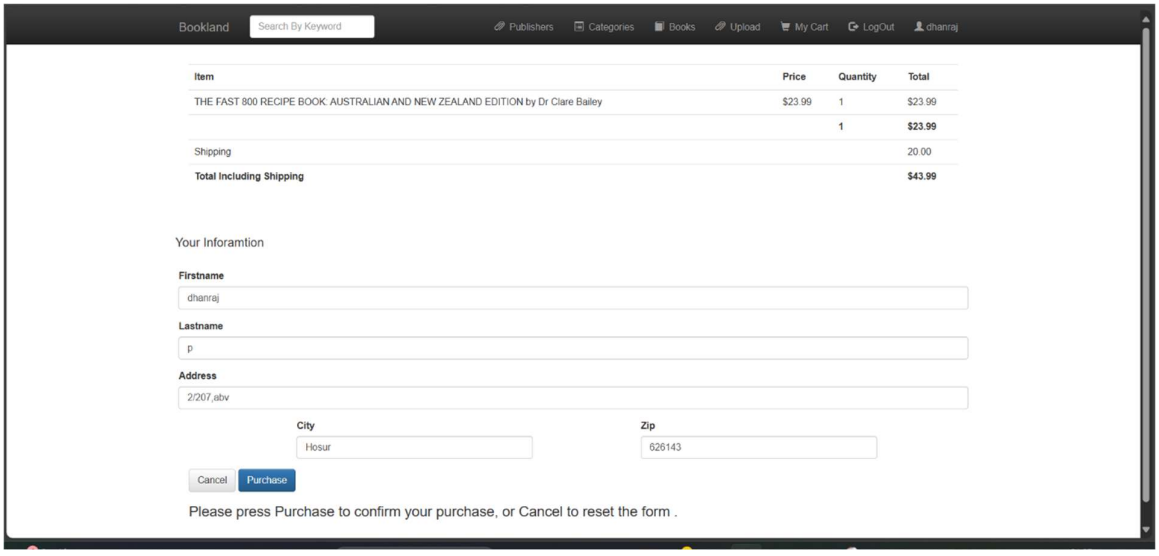
- 1 Arts And Entertainments
- 4 Children
- 1 Cooking,Food And Drink
- 0 Education
- 3 Fiction
- 1 Health And Diet
- 1 Non-Fiction
- List full of books

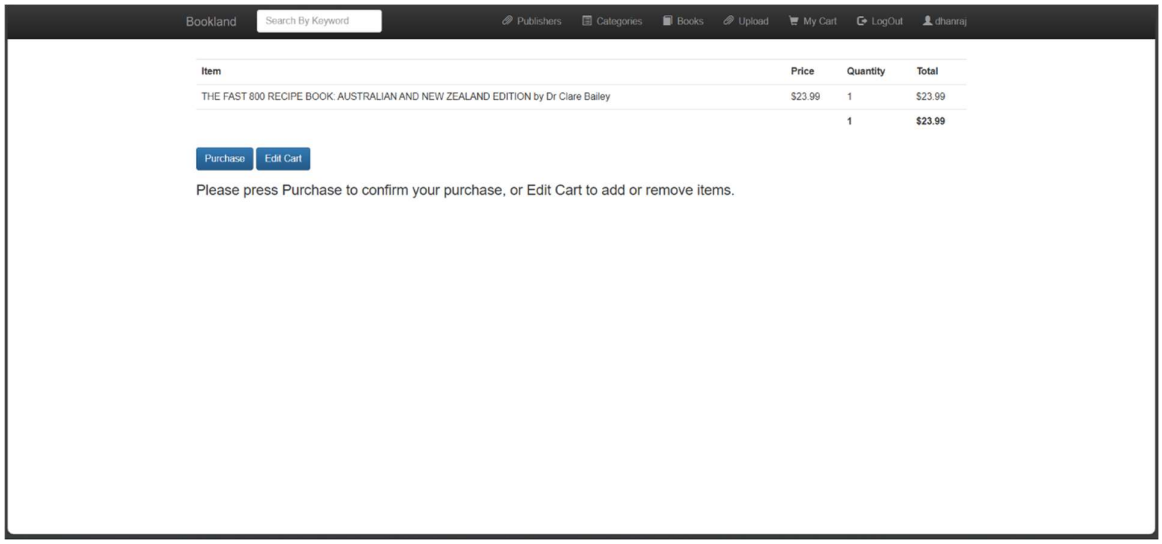

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
Categories > Children
KATHERINE RUNDELL
THE GOOD THIEVES
THE GOOD THIEVES
Get Details
ATLAS OF MONSTERS AND GHOSTS
ATLAS OF MONSTERS AND GHOSTS
Get Details

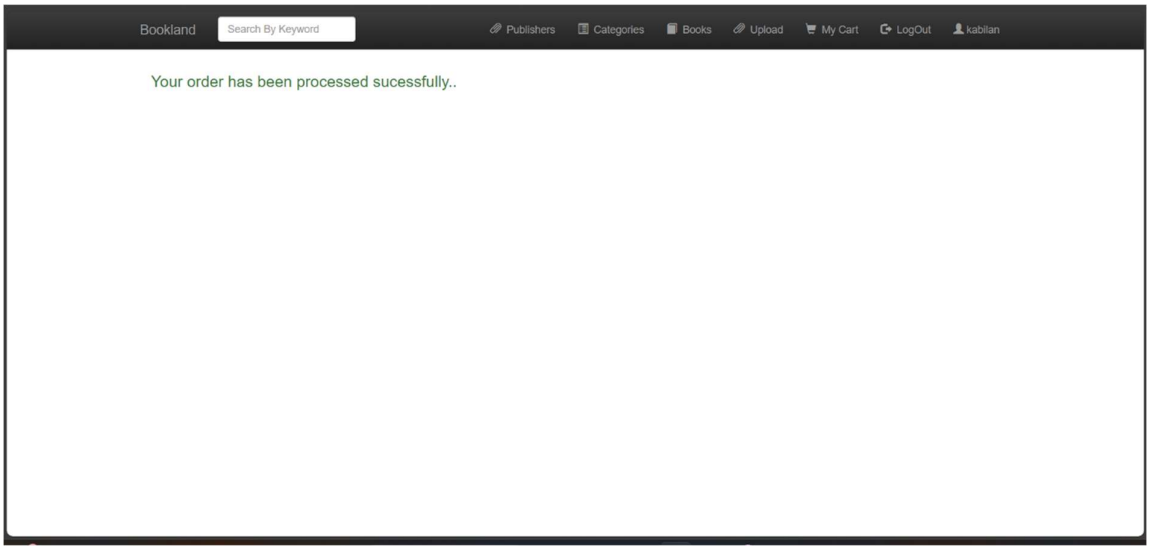

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
ISBN
Type
Resale
Title
Author
Image
Choose File
No file chosen
Description
Price
Publisher
Category
Add new book
Cancel

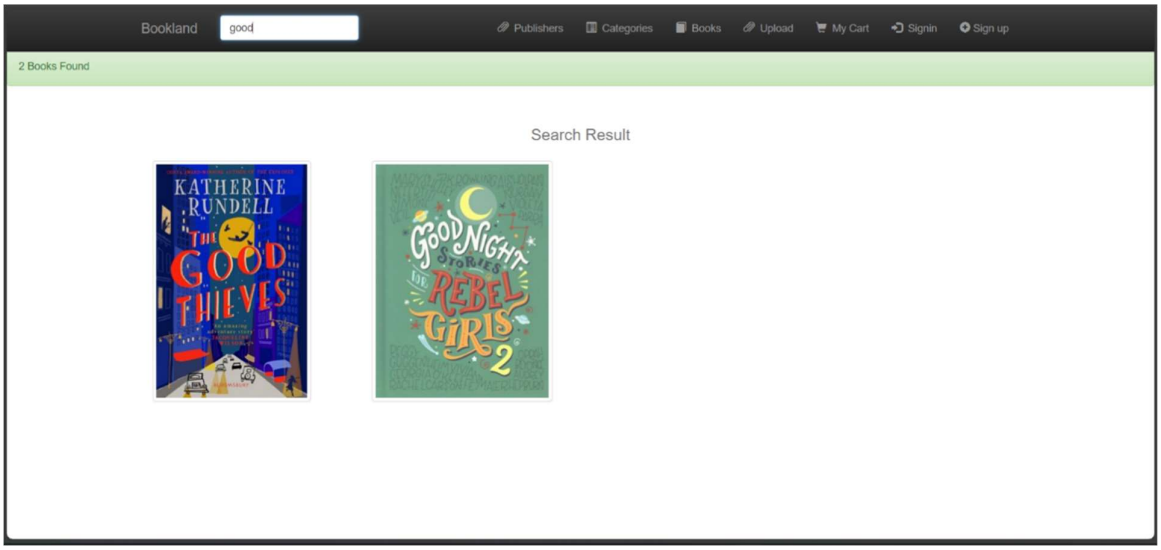

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
Books > THE FAST 800 RECIPE BOOK: AUSTRALIAN AND NEW ZEALAND EDITION
COMPANION TO THE NO.1 BESTSELLING THE FAST 800
the fast 800 recipe book
DR CLARE BAILEY and JUSTINE PATTISON
FOREWORD BY DR MICHAEL MOSLEY
Book Description
The companion to the No.1 bestseller The Fast 800. 150 delicious new recipes to help you combine rapid weight loss and intermittent fasting for long term good health. Foreword by Dr Michael Mosley. This companion cookbook to the international bestseller The Fast 800 by Dr Michael Mosley is filled with delicious, easy, low carb recipes and essential weekly meal planners, all carefully formulated by Dr Clare Bailey and Justine Pattison to help you lose weight, improve mood and reduce blood pressure, inflammation and blood sugars. Studies show that 800 calories is the magic number when it comes to successful dieting. Its high enough to be manageable, but low enough to speed weight loss and trigger a range of positive metabolic changes. In The Fast 800, Dr Michael Mosley brought together all the latest science, including Time Restricted Eating, to create an easy-to-follow programme, and this collection of all-new recipes, all photographed in full colour, will help you achieve all your goals. Every recipe is also calorie coded and noted with nutrition metrics to help you on your path to long term health. This diet changed my life Denise Bach, aged 51
Book Details
ISBN
978-0-321-94786-4
type
Author
Dr Clare Bailey
Price
23.99
categoryid
1
Add to cart

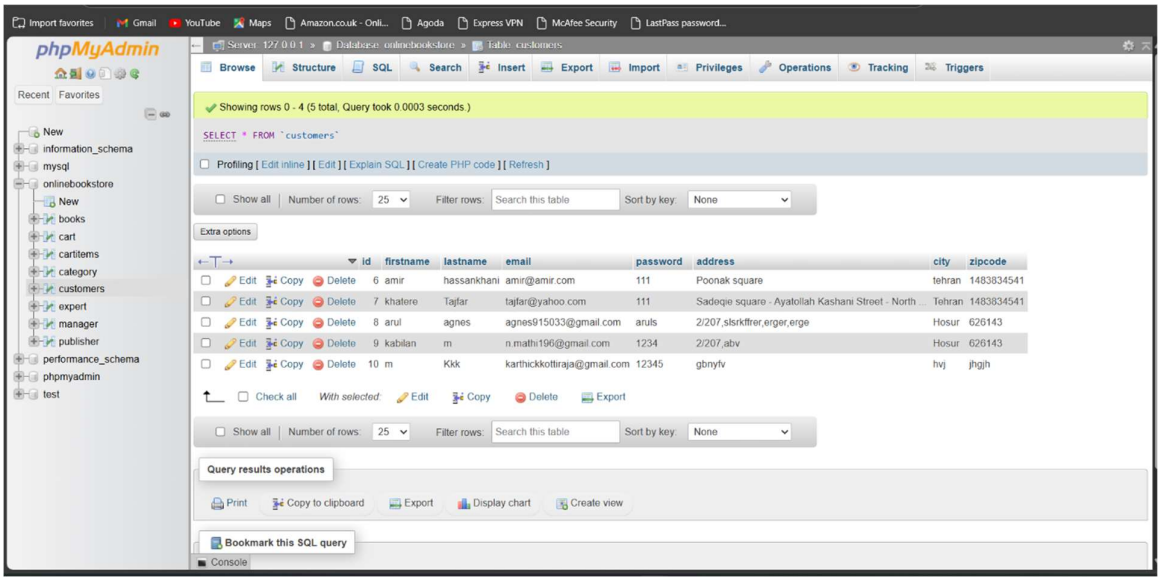

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
Item
Price
Quantity
Total
THE FAST 800 RECIPE BOOK: AUSTRALIAN AND NEW ZEALAND EDITION by Dr Clare Bailey
23.99
1
23.99
1
23.99
Save Changes
Go To Checkout
Continue Shopping

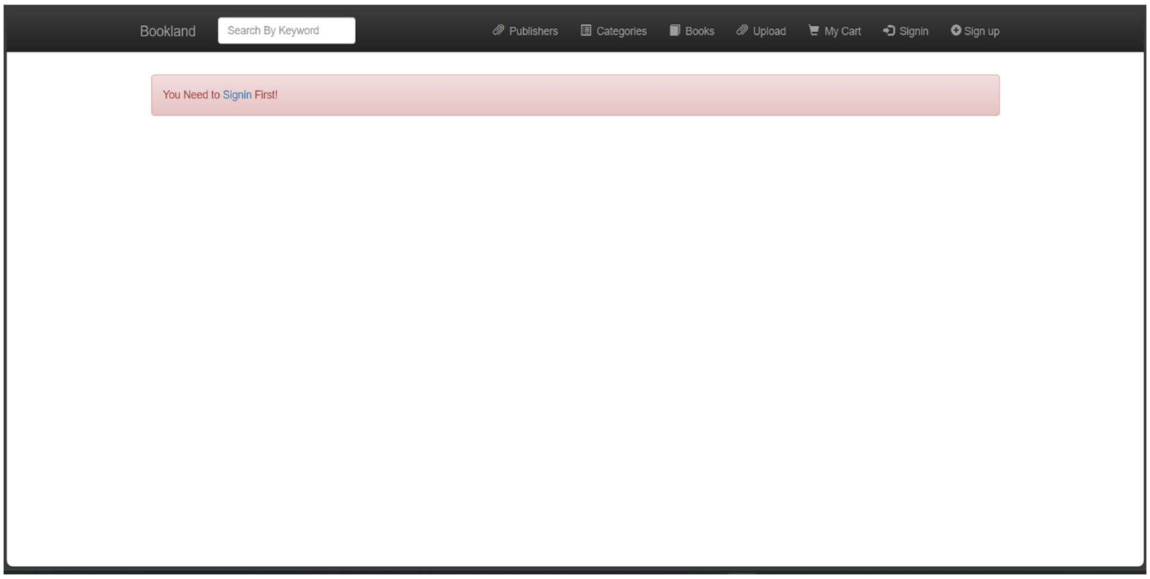

Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
Item
Price
Quantity
Total
THE FAST 800 RECIPE BOOK: AUSTRALIAN AND NEW ZEALAND EDITION by Dr Clare Bailey
$23.99
1
$23.99
1
$23.99
Shipping
20.00
Total Including Shipping
$43.99
Your Inforamtion
Firstname
dhanraj
Lastname
p
Address
2/207,abv
City
Hosur
Zip
626143
Cancel
Purchase
Please press Purchase to confirm your purchase, or Cancel to reset the form .


Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
dhanraj
Item
Price
Quantity
Total
THE FAST 800 RECIPE BOOK: AUSTRALIAN AND NEW ZEALAND EDITION by Dr Clare Bailey
$23.99
1
$23.99
1
$23.99
Purchase
Edit Cart
Please press Purchase to confirm your purchase, or Edit Cart to add or remove items.


Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
LogOut
kabilan
Your order has been processed sucessfully..


Bookland
good
Publishers
Categories
Books
Upload
My Cart
Signin
Sign up
2 Books Found
Search Result
KATHERINE RUNDELL
THE GOOD THIEVES
GOOD NIGHT STORIES FOR REBEL GIRLS 2


phpMyAdmin
Recent
Favorites
New
information_schema
mysql
onlinebookstore
New
books
cart
cartitems
category
customers
expert
manager
publisher
performance_schema
phpmyadmin
test
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)
SELECT * FROM `customers`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
id firstname lastname email password address city zipcode
Edit Copy Delete 6 amir hassankhani amir@amir.com 111 Poonak square tehran 1483834541
Edit Copy Delete 7 khatere Tajfar tajfar@yahoo.com 111 Sadeqie square - Ayatollah Kashani Street - North ... Tehran 1483834541
Edit Copy Delete 8 arul agnes agnes915033@gmail.com aruls 2/207,slsrkffrer,erger,erge Hosur 626143
Edit Copy Delete 9 kabilan m n.mathi196@gmail.com 1234 2/207,abv Hosur 626143
Edit Copy Delete 10 m Kkk karthickkottiraja@gmail.com 12345 gbnyfv hvj jhgjh
Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Console


Bookland
Search By Keyword
Publishers
Categories
Books
Upload
My Cart
Signin
Sign up
You Need to Signin First!

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The "Online Book Reselling and Donating" system has been developed to meet all specified requirements efficiently and effectively. The platform has been designed with simplicity and ease of use in mind, making it accessible for all types of users. It is a highly scalable solution, capable of handling increased loads as the user base grows, and offers a user-friendly interface that enhances the user experience.

The database has been carefully designed to offer flexibility, allowing for seamless integration and implementation. The system has been fully implemented and validated, with all phases of development guided by established methodologies to ensure quality and consistency. Even users with minimal training can quickly learn to navigate the system and generate reports as needed, promoting ease of access and usability.The software performs reliably, meeting all the primary goals of the project. Additionally, the system's architecture allows for future expansions or enhancements with only minor adjustments, making it adaptable to evolving requirements and additional features as needed.

6.2 FUTURE SCOPE

Software development is never–ending process and continues the life of the software as per the changing needs of the user from time to time. The project is no doubt has been developed keeping in mind easy modification and enhancement that may be required from time to time. However, there are many scopes to modify this software. As because due to shortage of time, we here become unable to include many things. We are trying to cover all their existing system for sales return records of the items but due to shortage of time we become unable to include many things. Due to lake of time I here include none

of them and a future scope one can develop these returns which are so much essential. Only with a little more doing it is possible to design the formats for those returns. Moreover, an on-line system will be more helpful to the organization. . With almost the same data with only a little modification an on-line system can be designed to fulfill their demands.

The database has been carefully designed to offer flexibility, allowing for seamless integration and implementation. The system has been fully implemented and validated, with all phases of development guided by established methodologies to ensure quality and consistency. Even users with minimal training can quickly learn to navigate the system and generate reports as needed, promoting ease of access and usability.

This will not only streamline the request process but also ensure that users can conveniently access real-time information on the available books, improving their overall experience. Meanwhile, the administrator will be able to handle requests and track user information seamlessly, reducing the time and effort previously required to manage these tasks manually.

APPENDICES

SOURCE CODE

Index.php

```
<?php
session_start();
$count = 0;
$title = "Index";
require_once "./template/header.php";
require_once "./functions/database_functions.php";
$conn = db_connect();
$row = select4LatestBook($conn);
?>
<br/> <br/>
<p class="lead text-center text-muted">OUR LATEST BOOKS</p>
<br><br>
<div class="row">
<?php foreach($row as $book) { ?>
<div class="col-md-3">
<a href="book.php?bookisbn=<?php echo $book['book_isbn']; ?>">
<img class="img-responsive img-thumbnail" src="./bootstrap/img/<?php echo
$book['book_image']; ?>">
</a>
</div>
<?php } ?>
</div>
<?php
if(isset($conn)) {mysqli_close($conn);}
require_once "./template/footer.php";
?>
```

Book.php

```
<?php
 session_start();
 $book_isbn = $_GET['bookisbn'];
 // connec to database
 require_once "./functions/database_functions.php";
 $conn = db_connect();
 $query = "SELECT * FROM books WHERE book_isbn = '$book_isbn'";
 $result = mysqli_query($conn, $query);
 if(!$result){
  echo "Can't retrieve data " . mysqli_error($conn);
  exit;
 }
 $row = mysqli_fetch_assoc($result);
 if(!$row){
  echo "Empty book";
  exit;
}
$title = $row['book_title'];
require "./template/header.php";
?>
<!-- Example row of columns -->
<p class="lead" style="margin: 25px 0"><a href="books.php">Books</a> > <?php echo $row['book_title']; ?></p>
<div class="row">
<div class="col-md-3 text-center"><img class="img-responsive img-thumbnail" src="./bootstrap/img/<?php echo $row['book_image']; ?>">
</div>
<div class="col-md-6">
<h4>Book Description</h4>
<p><?php echo $row['book_descr']; ?></p>
```

```
<h4>Book Details</h4>
<table class="table">
<?php foreach($row as $key => $value){
if($key == "book_descr" || $key == "book_image" || $key == "publisherid" || $key == "book_title"){continue;
}
switch($key){
case "book_isbn":
$key = "ISBN";
break;
case "book_title":
$key = "Title";
break;
case "book_author":
$key = "Author";
break;
case "book_price":
$key = "Price";
break;
}
?>
<tr>
<td><?php echo $key; ?></td>
<td><?php echo $value; ?></td>
</tr>
<?php
}
if(isset($conn)) {mysqli_close($conn); }
?>
</table>
<form method="post" action="cart.php">
```

```
<input type="hidden" name="bookisbn" value="<?php echo $book_isbn;?>">
<input type="submit" value="Add to cart" name="cart" class="btn btn-primary">
</form>
</div>
</div>
<?php
require "./template/footer.php";
?>
```

Signup.PHP

```
<?php
    $title = "User SignUp";
    require_once "./template/header.php";
?>
    <form class="form-horizontal" method="post" action="user_signup.php">
    <div class="form-group">
   <label for="exampleInputEmail1">Firstname</label>
 <input type="text" class="form-control" aria-describedby="emailHelp" placeholder="Firstname" name="firstname">
</div>
<div class="form-group">
<label for="exampleInputEmail1">Lastname</label>
<input type="text" class="form-control" aria-describedby="emailHelp" placeholder="Lastname" name="lastname">
 </div>
<div class="form-group">
<label for="inputEmail4">Email</label>
<input type="text" class="form-control" id="inputEmail4" placeholder="Email" name="email">
</div>
<div class="form-group">
<label for="inputPassword4">Password</label>
<input type="password" class="form-control" id="inputPassword4" placeholder="Password" name="password">
```

```
</div>
<div class="form-group">
<label for="inputAddress">Address</label>
<input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St" name="address">
</div>
<div class="form-row">
<div class="form-group col-md-4">
<label for="inputCity">City</label>
<input type="text" class="form-control" id="inputCity" name="city">
 </div>
<div class="form-group col-md-2">
</div>
<div class="form-group col-md-4">
<label for="inputZip">Zip</label>
<input type="text" class="form-control" id="inputZip" name="zipcode">
</div>
</div>
<div class="form-group col-md-12">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
<div style="position:fixed; bottom:120px">
<?php
$fullurl="http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
if(strpos($fullurl,"signup=empty")==true){
echo '<P style="color:red">You did not fill in all the fields.</P>';
exit();
}
if(strpos($fullurl,"signup=invalidemail")==true){
echo '<P style="color:red">You did not enter a valid email address.</P>';
exit();
}
```

```
?>
</div>
<?php
   require_once "./template/footer.php";
?>
```

Email.PHP

```
<?php
require_once('SMTP.php');
require_once('PHPMailer.php');
require_once('Exception.php');
use \PHPMailer\PHPMailer\PHPMailer;
use \PHPMailer\PHPMailer\Exception;
$mail=new PHPMailer(true); // Passing `true` enables exceptions
try {
//settings
$mail->SMTPDebug=2; // Enable verbose debug output
$mail->isSMTP(); // Set mailer to use SMTP
$mail->Host='smtp.gmail.com';
$mail->SMTPAuth=true; // Enable SMTP authentication
$mail->Username='donatingnest@gmail.com'; // SMTP username
$mail->Password='swa2510@'; // SMTP password
$mail->SMTPSecure='ssl';
7$mail->Port=465;
$mail->setFrom('support@bookland.com', 'bookland');
recipient
$mail->addAddress('swathimathi5@gmail.com','SWATHI');     // Add a recipient
//content
$mail->isHTML(true); // Set email format to HTML
$mail->Subject='Order Confrm';
$mail->Body='Hi, Your ordr has been sucessfully Placed';
$mail->AltBody='This is the body in plain text for non-HTML mail clients';
```

```
$mail->send();
$mail->SMTPSecure='ssl';
7$mail->Port=465;
$mail->setFrom('support@bookland.com', 'bookland');
recipient
$mail->addAddress('swathimathi5@gmail.com','SWATHI');     // Add a recipient
//content
$mail->isHTML(true); // Set email format to HTML
$mail->Subject='Order Confrm';
$mail->Body='Hi, Your ordr has been sucessfully Placed';
$mail->AltBody='This is the body in plain text for non-HTML mail clients';
header('location: index.php');
}
catch(Exception $e) {
echo 'Message could not be sent.';
echo 'Mailer Error: '.$mail->ErrorInfo;
}
?>
```

REFERENCES

[1]. Ansari Sharim Iqbal, Ansari Sharique Taj, Naik Owais Anwar, Mr. Kalidas Bhawale. "Online Book Reselling and Donation", International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0056(Online), Volume-6 Issue-4, Apr 2019.

[2]. Gyanchand Dhiwar, Avinash Kaushik, Darshan Bareth, Kritik Kumar Gadheval, Prof.Vivek Kumar Sinha., "Second hand online book store (E-commerce website)," International Journal Of Research Publication and Reviews(IJRPR), ISSN: 2582-7421 vol. 4, June 2023.

[3]. Riya Kalburg, Mohammed Faiz Sayyed. “Books-Are-Us: Book Donation Website”, International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0058, Volume-6 Issue-2, Feb 2021.

[4]. Paul Kipling, T.D. Wilson. “Publishing, bookselling and the World Wide Web” , ResearchGate, 2023.

[5]. Trish Groves, Donating books and journals to less developed countries, ERIC(Education Resources Information center), February 2023.