

GOVERNMRNT COLLEGE UNIVERSITY FAISALABAD



Medical Books E-Store

By

Noraiz Latif

2017-GCUF-061351

Muhammad Numan

2017-GCUF-061374

Project Submitted in Partial Contentment of the Requirements for the Degree of

BACHELOR OF SCIENCE

IN

INFORMATION TECHNOLOGY



Department of Computer Science and Information Technology

Government Graduate College Samanabad Faisalabad

2021

GOVERNMENT COLLEGE UNIVERSITY FAISALABAD



Medical Books E-Store

By

Noraiz Latif

2017-GCUF-061351

Muhammad Numan

2017-GCUF-061374

Project Submitted in Partial Contentment of the Requirements for the Degree of

BACHELOR OF SCIENCE

IN

INFORMATION TECHNOLOGY



Department of Computer Science and Information Technology

Government Graduate College Samanabad Faisalabad

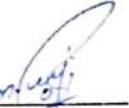
2021

CERTIFICATE

This is to certify that **Noraiz Latif** bearing Registration No. **2017-GCUF-061351** and **Muhammad Numan** bearing Registration No. **2017-GCUF-061374** have completed the final project titled "**Medical Books E-Store**" at the **Department of Information Technology, Govt. Graduate College Samanabad Faisalabad**, to fulfill the partial requirement for the degree of BS - IT.

Supervisor

Muhammad Majid

Signature: 

Committee Members

Muhammad Majid

Signature: 

Umair Yousaf

Signature: _____

Hina Zafar

Signature: _____

Kinza Nadeem

Signature: _____

Project Coordinator

Head of Department

Principal

DECLARATION

The work reported in this project was carried by me under the supervision of Project Supervisor, **Muhammad Mafid**, at Government College Samanabad Faisalabad.

We hereby declare that this project and the contents of the project are the product of my effort.

We further declare that this work has not been submitted for the award of any other degree.

The institution may take action if the provided information is found inaccurate at any stage.

Name : Noraiz Latif
Registration No. : 2017-GCUF-061351

Name : Muhammad Numan
Registration No. : 2017-GCUF-061374

ACKNOWLEDGEMENT

In the name of **ALMIGHTY ALLAH**, the most merciful, the most beneficent. Who is the Lord and Owner of all authorities and capabilities who endowed his greatness on us? He is the entire source of knowledge of wisdom and who blessed us with the ability to do work. We are grateful to the **prophet Muhammad (PBUH)** who gave us the spirit to learn. No doubt in it because of **ALMIGHTY ALLAH** that today we can complete our project.

We are also thankful to our honorable Supervisor **Muhammad Majid**, Department of Information Technology of Government College University Faisalabad, for his guidance, encouragement, support, and admirable help. He always encouraged us to come to our project work. In fact, without his efforts and guidance, we may not be able to complete our work.

We should like to thank all the faculty of Information Technology of Government Graduate College Samanabad Faisalabad for this great support and help for proper completion of our project work. We should feel it necessary to express our exclusive love and adoration to our parents and other family members for their support, encouragement, and tremendous contribution to our project completion.

May all they live long to see our dreams being fulfilled. Ameen!

With Best Regards,

Noraiz Latif

Muhammad Numan

ABSTRACT

An online medical bookstore software project that acts as a central database containing various books which are helpful for medical students in stock along with their title, author, and cost. This project is a website that acts as a central book store. This web project is developed using HTML, CSS, and BOOTSTRAP as the front end and SQL as the back end. The SQL database stores various book-related details. A user visiting the website can see a wide range of books arranged in respective categories. The user may select the desired book and view its price. The user may even search for specific books on the website. Once the user selects a book, he then has to fill in a form and the book is booked for the user. Online Book Purchase and Sales system is an open-source web-based project developed with PHP scripting language and an open-source MYSQL database server. This web application provides a huge database of medical books, where any medical student can search for interesting books and purchase them through different payment options where users can send a purchase order to a merchant online. Users can register themselves in the index webpage through the registration link as a member or admin where they can add their books to our database for sale. The optimization of this PHP project has been done and works successfully without delay at the server. To make our web site more attractive and user friendly, we introduced some more features like.

Table of Contents

Chapter 1	INTRODUCTION	1
1.1 Purpose of Project.....	1	
1.2 How it works.....	1	
1.3 Project Scope	1	
1.3.1 Cost.....	1	
1.3.2 Effort.....	1	
1.3.3 Time.....	2	
1.4 Project Planning.....	2	
1.5 Project Modules	3	
1.5.1 Main Screen.....	3	
1.5.2 Signup and Login	3	
1.5.3 Payment Method	3	
1.5.4 Cart	3	
1.5.5 User	3	
1.5.6 Admin.....	4	
1.5.7 Report	4	
Chapter 2 BACKGROUND and PROBLEM DEFINITION	5	
2.1 Background Research.....	5	
2.2 Existing Technology.....	5	
2.3 Area of Study.....	5	
2.3.1 Specifications.....	5	
2.3.1 Tools	5	
2.4 Reason for the Project	6	
2.5 Objective of the Project	6	
2.6 Methodology	6	
Chapter 3 SOFTWARE REQUIREMENT ANALYSIS.....	9	
3.1 System Requirement Specification	9	
3.2 Function requirements	9	
3.2.1 Functional requirements	10	
3.3 Non-functional requirements	10	
3.3.1 User Friendly	10	
3.3.2 Graphical User Interface.....	10	
3.3.3 Performance	10	
3.3.4 Availability	11	
3.3.6 Privacy	11	

3.3.7 Quality.....	11
3.3.8 Security Requirements	11
3.4 Interface Specifications	11
3.5 Software and Hardware Requirements	11
3.5.1 Software Requirements	11
3.5.2 Hardware Requirements.....	12
Chapter 4 SYSTEM DESIGN.....	13
4.1 Use Case Diagram.....	13
4.1.1 Use Case Search Book	14
4.1.2 Use Case Sign up	14
4.1.3 Use Case Login	15
4.2 System Sequence Diagram	16
4.2.1 User Side	16
4.2.2 Admin Side	17
4.4 ER Diagram.....	18
4.4 ER Diagram	19
4.5 Data Flow Diagram.....	20
4.5.1 DFD Level 0	20
4.5.2 DFD Level 1 User Side.....	20
4.5.3 DFD Level 1 Admin Side.....	21
4.5.4 DFD Level 2 Admin Side (A)	21
4.5.4 DFD Level 2 Admin Side (B).....	22
Chapter 5 TESTING AND IMPLEMENTATION	23
5.1 Testing Techniques	24
5.1.1 Unit Testing	24
5.1.2 Integration Testing.....	25
5.1.3 System Testing.....	25
5.1.4 Acceptance Testing	25
5.1.5 Black Box Testing	25
5.1.6 White Box Testing.....	25
5.1.7 Grey Box Testing.....	25
5.2 Test Approach	26
5.2.1 Bottom-up Approach.....	26
5.2.2 Top-down approach.....	26
5.3 Test Cases	26
TC - 1: Start website and Signup.....	27

List of Tables

Table 1 Use Case Search Book	14
Table 2 Use Case Sign Up	14
Table 3 Use Case Login.....	15
Table 4 Start Website and Sign Up	27
Table 5 Sign Up Using Incorrect Information.....	28
Table 6 Anyone can send feedback about the website	29
Table 7 Login into Website	30
Table 8 Add to Cart	31
Table 9 Add to Cart Without Login	32
Table 10 Edit Cart.....	33
Table 11 Upload a new or old Book.....	34
Table 12 Delete a new or old Book.....	35
Table 13 Create a book Category.....	36
Table 14 Delete a book Category.....	37
Table 15 Manage Orders.....	38
Table 16 Generate Reports.....	39

List of Figures

Figure 1 Software Development Life Cycle	3
Figure 2 Waterfall Model	7
Figure 3 Use Case Diagram	13
Figure 4 System Sequence Diagram for User	16
Figure 5 System Sequence Diagram for Admin	17
Figure 6 Class Diagram	18
Figure 7 ER Diagram	19
Figure 8 DFD Level 0	20
Figure 9 DFD Level 1 User Side	20
Figure 10 DFD Level 1 Admin Side	21
Figure 11 DFD Level 2 Admin Side A	21
Figure 12 DFD Level 2 Admin Side B	22
Figure 13 Home Page	41
Figure 14 Contact Us	42
Figure 15 Sign Up Page	43
Figure 16 Login Page	44
Figure 17 Search a Book	45
Figure 18 Book Description	46
Figure 19 Add to Cart	46
Figure 20 Check Out	47
Figure 21 Payment	48
Figure 22 Admin Login Page	49
Figure 23 Admin Dashboard	50
Figure 24 Manage Books	51

Chapter 1

INTRODUCTION

1.1 Purpose of Project

This project is like an E-Bookstore website where books can be bought from the comfort of home through the Internet. An online bookstore is a virtual store on the Internet where customers can browse the catalog and select books of interest related to the medical field. Users can select many books and those books stored in a cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the students will be asked to fill in the basic details or select a billing address, a shipping address, and payment information such as Jazz cash.

1.2 How it works

This is a website where medical students can sell and buy old books as well as buy new books for their studies. Students can get access by providing their basic information like billing address, shipping address and can pay through Jazz cash.

1.3 Project Scope

The scope of this project is to develop a web-based application that helps buyers instantly find and connect with thousands of booksellers in one place. This website will provide a platform to buyers on their smartphones/desktops for finding the books of their desired subject according to their requirements. Finally, the website will allow the users to rate and give their reviews about any seller and the services they provide which will make this website more useful for the users.

1.3.1 Cost

The cost required in the purposed system is comparatively higher than the existing systems. Because the previous system works manually.

1.3.2 Effort

Compared to the existing system, the proposed system will provide a better work environment in which there will be cases of work and the effort required will be comparatively less than the existing system.

1.3.3 Time

The time required to selling and buying old and new books will be comparatively less than the existing systems.

1.4 Project Planning

The purpose of this project is to sell and buy old books and medical students will be able to buy new books online.

This project is supported by Noraiz Latif and Muhammad Numan and the Supervisor Muhammad Majid our IT staff Member has supported us in fulfilling the task of this project. They motivated us to do this project. The supervisor helped us to develop this project more efficiently. Then we start to develop this project.

The goal of this section is to provide a set of recommendations that will help you plan appropriately for a successful project. In this section, we use the life cycle model employed broadly at Microsoft. This model is a combination of iterative and waterfall life cycle models. In this model, there are five phases whose boundaries define a sequential set of milestones for the project. The phases, in order of execution, are as follows:

- **Requirements:** To make an atmosphere-friendly and user-friendly books selling website through which medical students can sell and buy old bold books and also able to buy new books.
- **Design:** Based on the functional requirements, physical design specifications are created and prototyping is conducted to verify design ideas and investigate the capabilities.
- **Implementation:** Using the design and functional specifications, the coding is done.
- **Verification:** This is the process of testing the product to verify that it performs according to the specifications.
- **Release:** After the product has been fully verified it is packed and prepared for release to Customers.

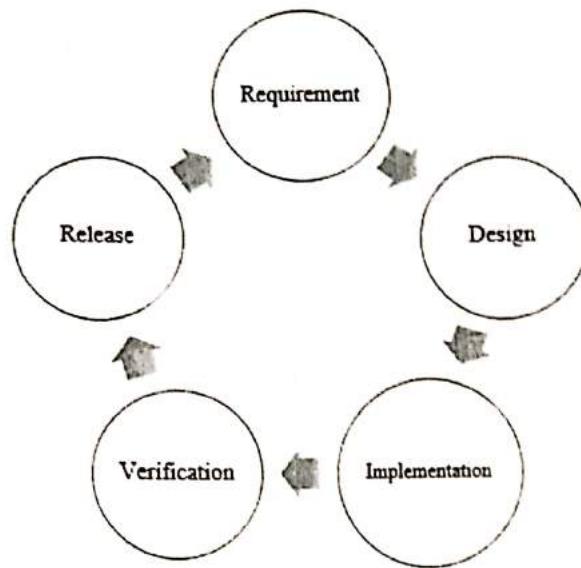


Figure 1 Software Development Life Cycle

1.5 Project Modules

1.5.1 Main Screen

On this screen, users and admin can access their dashboard.

1.5.2 Signup and Login

The users can sign up and log in. The sign-up will be done using correct information. The sign-in will be done using "email" and "password".

1.5.3 Payment Method

We have added a payment method so that payment can be paid online.

1.5.4 Cart

The User can add a book into the cart and then checkout for it so that the user can proceed with payment

1.5.5 User

We have added a user module so that sellers and buyers can easily access their accounts.

1.5.6 Admin

The admin can add new books and also have a right to edit and delete books of users that are added by users.

1.5.7 Report

The admin can add new tables, add new dishes, see existing orders, see feedback, and can generate reports.

Chapter 2 BACKGROUND and PROBLEM DEFINITION

2.1 Background Research

There are many online book stores like Powell's, Amazon which were designed using Html. We want to develop a similar website using Html, CSS, Bootstrap, Php.

Online Book store is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add it to the shopping cart, and finally purchase using Jazz Cash. The user can log in using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number, and shipping address.

2.2 Existing Technology

Now a day's students have to go to the market to buy books related to the medical field.

Sometimes students cannot get their desired books from their cities, which creates a problem for students while studying.

2.3 Area of Study

This section is related to the description of the system specifications and the tools that were used to design and develop the project.

2.3.1 Specifications

This is a software-based system and definitely, the system will run through a web browser which must be able to run the system.

2.3.1 Tools

We have used many tools which we had used during the designing and development of the system like:

- Html
- CSS
- Bootstrap
- Php
- Java Script

- Web Browser
- Xampp Server
- Ms. Word

2.4 Reason for the Project

This project is like an E-Bookstore website where books can be bought from the comfort of home through the Internet. An online bookstore is a virtual store on the Internet where customers can browse the catalog and select books of interest related to the medical field. Users can select many books and those books stored in a cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the students will be asked to fill in the basic details or select a billing address, a shipping address, and payment information such as Jazz cash.

2.5 Objective of the Project

The main objective of the project is to create an online book store especially for medical students which will allow users to search and purchase a book based on the title. The selected books are displayed in the cart and the user can order their books online through the Jazz Cash payment method. The Administrator will have additional functionalities when compared to the common user.

2.6 Methodology

We have used the “Waterfall Model” for the development of our project. Waterfall Model is a sequential model that divides software development into pre-defined phases. Each phase must be completed before the next phase can begin with no overlap between the phases. Each phase is designed for performing specific activities during the SDLC phase. It was introduced in 1970 by Winston Royce.

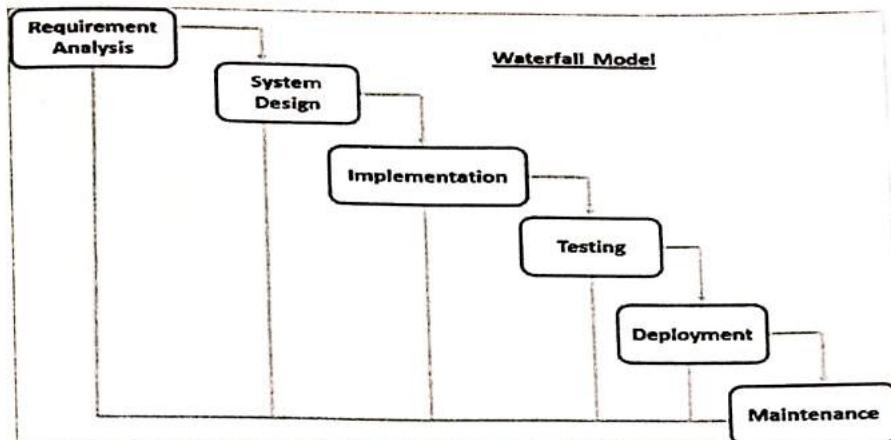


Figure 2 Waterfall Model

The sequential phases in the Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design:** The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation:** With inputs from the system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of the system:** Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance:** Some issues come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions

are released. Maintenance is done to deliver these changes in the customer environment.

Chapter 3 SOFTWARE REQUIREMENT ANALYSIS

3.1 System Requirement Specification

System Requirements Specification (abbreviated SRS when need to be distinct from a Software Requirements Specification SRS) is a structured collection of information that embodies the requirements of a system.

A business analyst sometimes titled a system analyst is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and requirements that are required for the project development. To derive the requirements, we need to have a clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

3.2 Function requirements

A functional requirement is **describing the behavior of the system** as it relates to the system's functionality.

Functional requirements are what the user expects from the software for example if the application is a bank application that application should be able to create a new account, update the account, delete an account, etc. functional requirements are detailed and are specified in the system design. **Functional Requirements:** specify the **functionality** of the system. (E.g. fields in a form).

A functional requirement describes *what a software system should do*. Let me elaborate. An example of a functional requirement would be that a system must send an email whenever a certain condition is met (e.g. an order is placed, a customer's sign up).

3.2.1 Functional requirements

- The system should be able to get the user name from the user.
- The system should be able to get the correct password for authentication from the user.
- The system should be able to log in.
- The system should be able to show the main screen.
- The system should be able to differentiate user types.
- The system should be able to show the profile.
- The system should be able to upload book pictures.

3.3 Non-functional requirements

Typically, non-functional requirements fall into areas such as Non-functional requirements, these come in two types:

- Performance constraints – what performance is required from the system e.g. It will update all customer records overnight.
- Development constraints – what restrictions on development will apply e.g. the system must be available by a certain date.

3.3.1 User Friendly

This website needs to be user-friendly when using its user interface.

3.3.2 Graphical User Interface

By using GUI's, it should make the application more user-friendly and better to use instead of a command line. Buttons will be used. For example: "Modified data in a database should be updated for all users access it within few seconds."

3.3.3 Performance

The website gives a quick response to the user when he tries to perform any type of function.

3.3.4 Availability

The system should be available at any time of use so that the users can use it at any time without any difficulty

3.3.6 Privacy

We provide complete privacy to the contractors. Everything in the project is in the privacy between the Patient and us. There is no problem with privacy for the Patient.

3.3.7 Quality

The quality of the project is so good everything in its full fill all the requirements of the Owner of the website and the process is done in a good manner.

3.3.8 Security Requirements

It needs an authentic username and password to secure our system privacy. If anyone wants to use it or hack it he/she will not allow using the system without an authentic user name and password.

3.4 Interface Specifications

The interface specification is a document that captures the details of the software user interface into a written document. The specification covers all possible actions that an end-user may perform and all visual, auditory, and other interactive elements.

3.4.1 Software Quality Attributes

- The Quality of the System is maintained in such a way, that the system is made user-friendly.
- The system quality attributes are assumed as under:
- Accurate and hence reliable.
- Secured.
- Fast speed.
- Compatibility.

3.5 Software and Hardware Requirements

3.5.1 Software Requirements

- Operating System : Windows 2000/XP/8.1/10
- Languages : HTML, PHP, CSS, JavaScript, JQuery
- Backend : MYSQL

- Browser : Opera, Mozilla Fire Fox, Google Chrome

3.5.2 Hardware Requirements

- PC : Pentium IV
- Processor : 1 GHz CPU
- RAM : 512 MB
- Hard Disk : 5 GB
- Internet : Compulsory

Chapter 4 SYSTEM DESIGN

4.1 Use Case Diagram

User and Administrator are the two actors included in the Online Book Store.

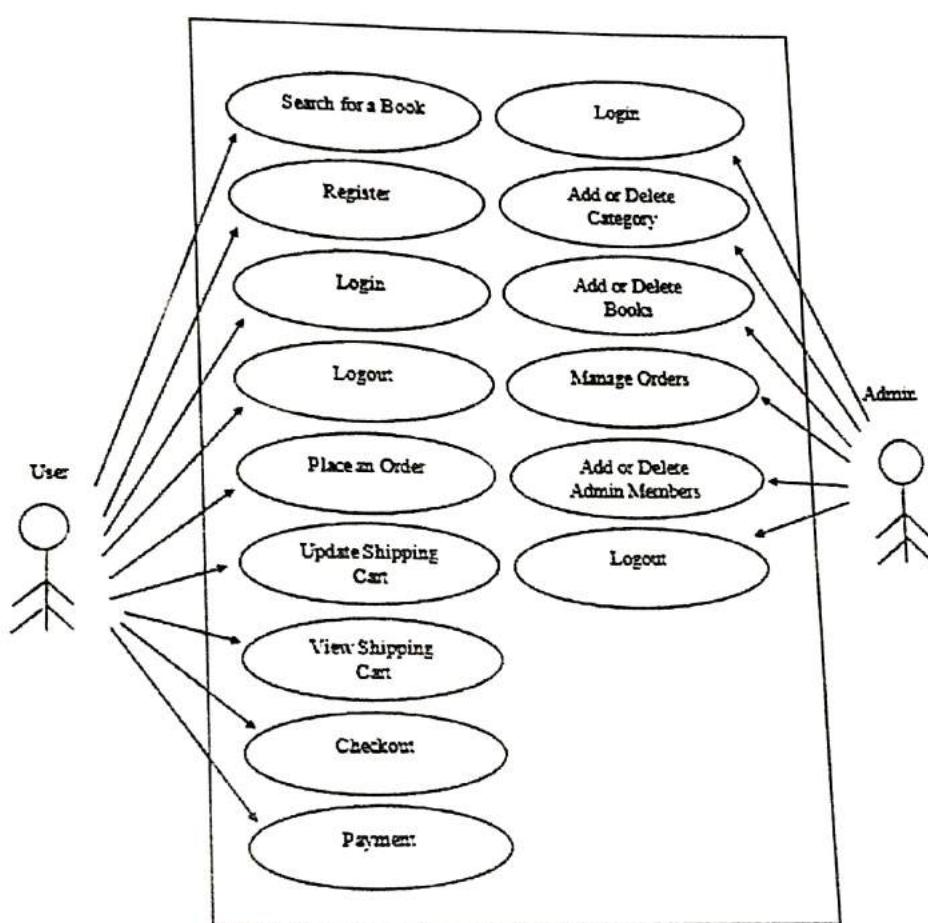


Figure 3 Use Case Diagram

4.1.1 Use Case Search Book

Table 1 Use Case Search Book

Use case No.	01
Use case name	Search Book
Actor	Admin, User
Pre-condition	N/A, Registered
Action	Enter required detail
Postcondition	Signed up
Exception	Incorrect format
Number of occurrences	Whenever needed

4.1.2 Use Case Sign up

Table 2 Use Case Sign Up

Use case No.	02
Use case name	Sign up
Actor	Admin, User
Pre-condition	N/A, Registered
Action	Enter required detail
Post condition	Signed up
Exception	Incorrect format
Number of occurrences	Only one time

4.1.3 Use Case Login

Table 3 Use Case Login

Use case No.	03
Use case name	Login
Actor	Admin, User
Pre-condition	Sign up
Action	Enter Email Address and Password
Post condition	Logged in
Exception	Incorrect format
Number of occurrences	Whenever needed

4.2 System Sequence Diagram

4.2.1 User Side

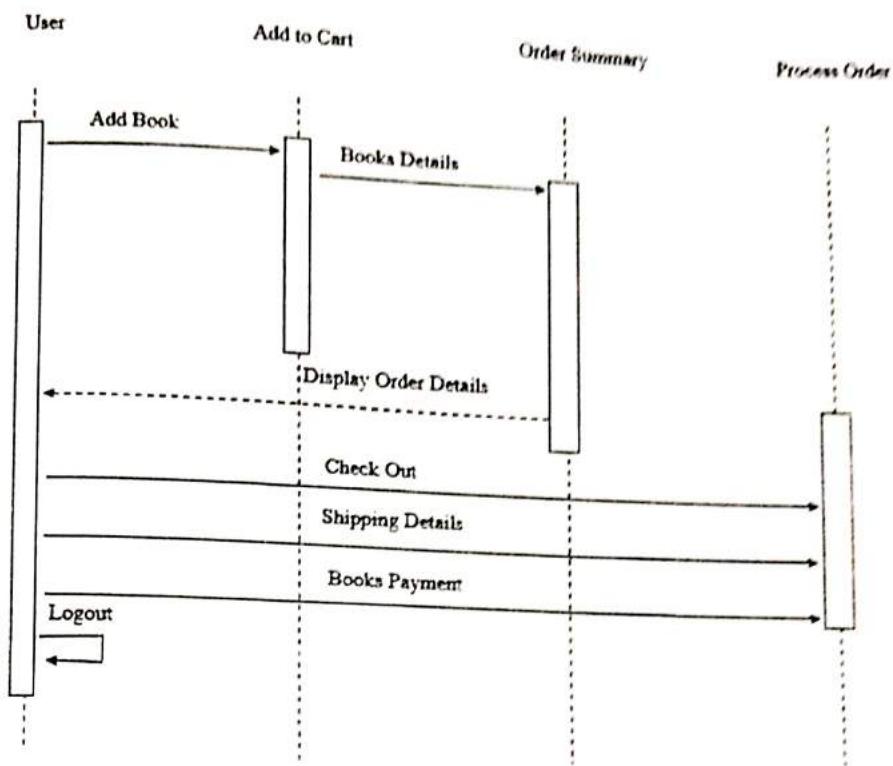


Figure 4 System Sequence Diagram for User

4.2.2 Admin Side

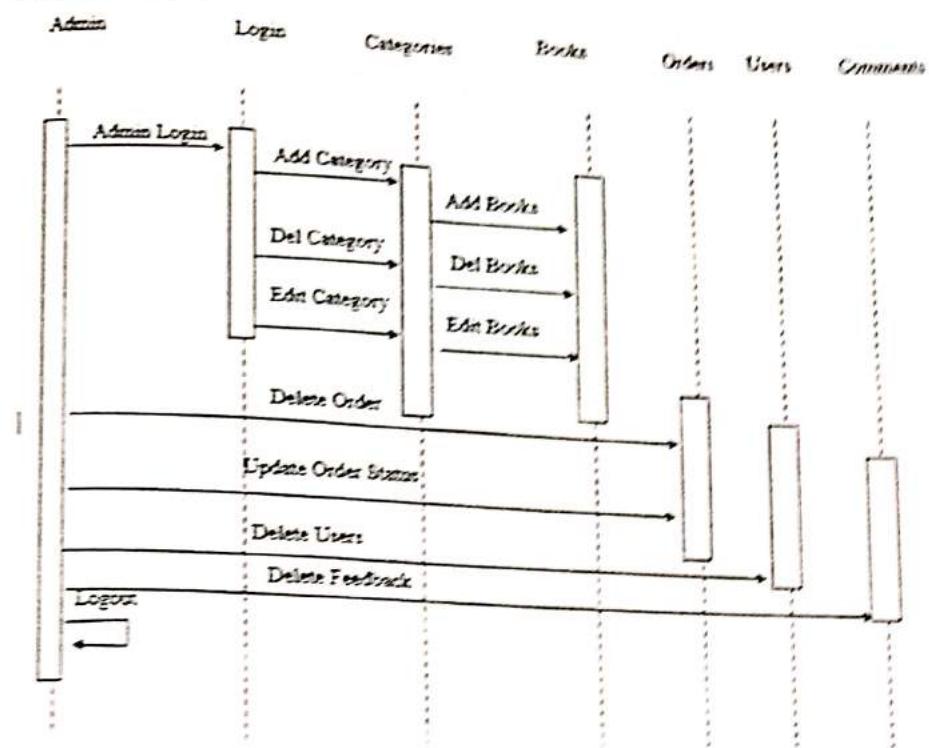


Figure 5 System Sequence Diagram for Admin

4.4 ER Diagram

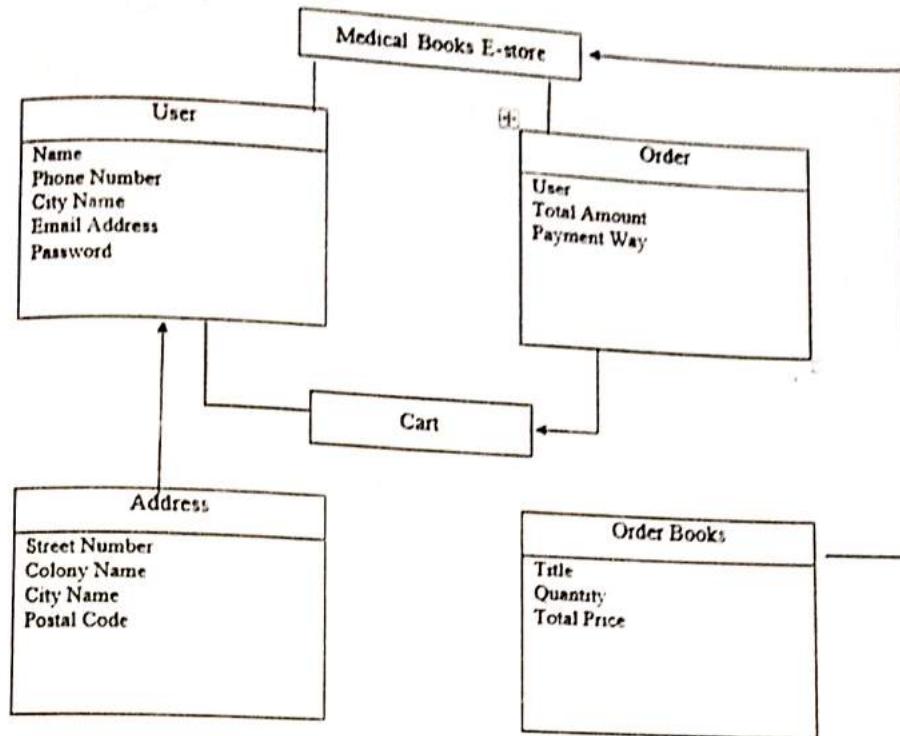


Figure 6 Class Diagram

4.4 ER Diagram

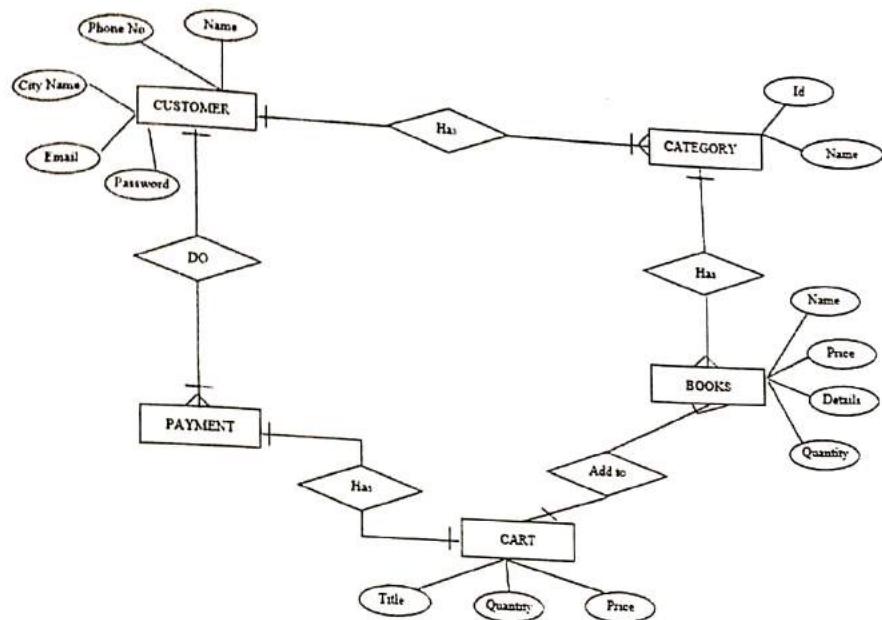


Figure 7 ER Diagram

4.5 Data Flow Diagram

4.5.1 DFD Level 0

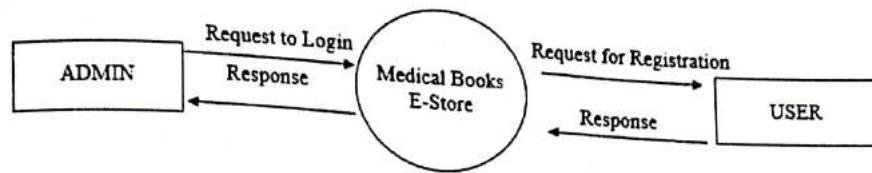


Figure 8 DFD Level 0

4.5.2 DFD Level 1 User Side

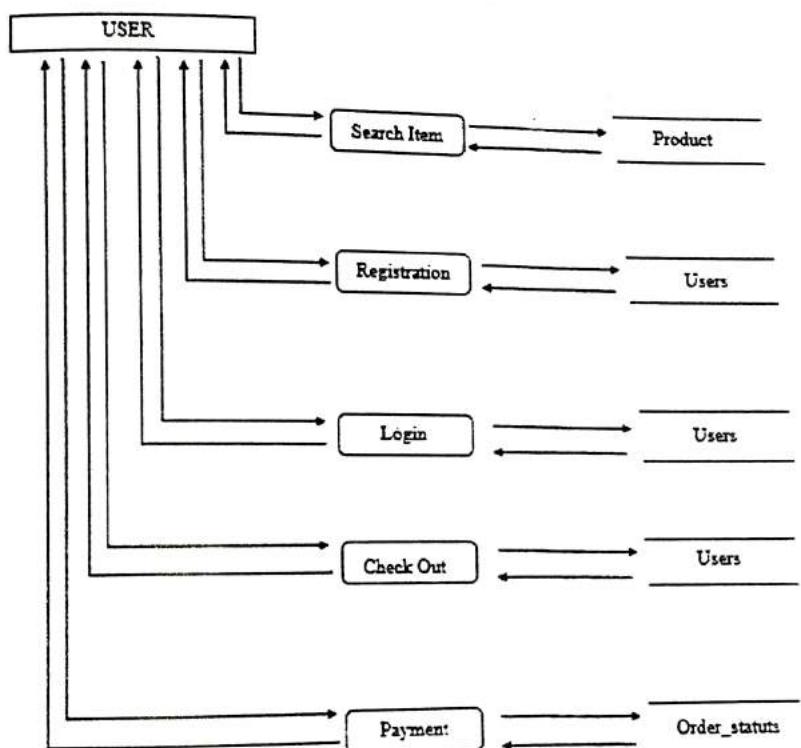


Figure 9 DFD Level 1 User Side

4.5.3 DFD Level 1 Admin Side

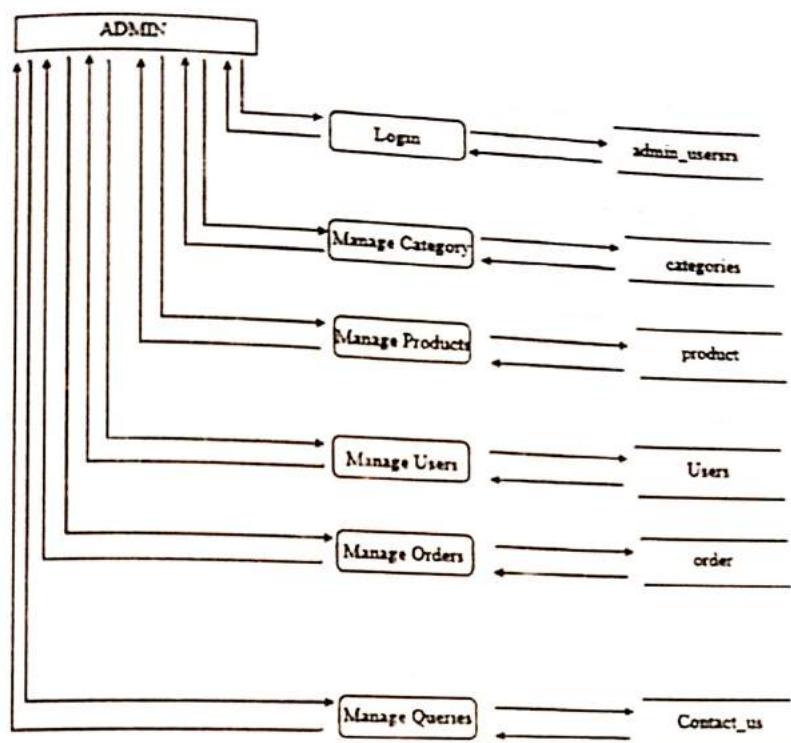


Figure 10 DFD Level 1 Admin Side

4.5.4 DFD Level 2 Admin Side (A)

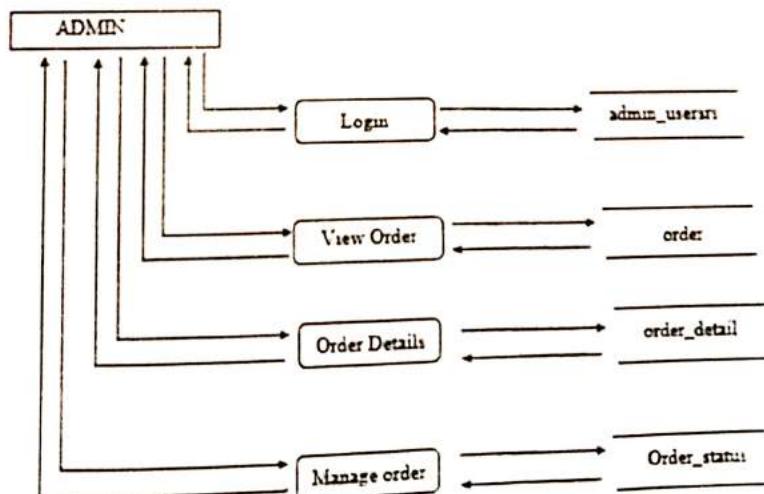


Figure 11 DFD Level 2 Admin Side A

4.5.4 DFD Level 2 Admin Side (B)

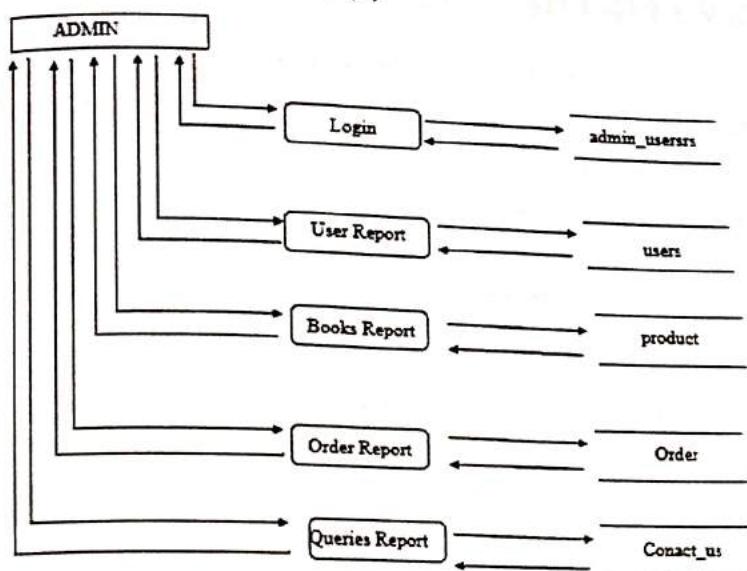


Figure 12 DFD Level 2 Admin Side B

Chapter 5

TESTING AND IMPLEMENTATION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. Testing is an internal part of any system or project. If a system is implemented without being tested it may lead to erroneous working and dissatisfaction on part of the customer. It will also prove disastrous to the reputation of the organization or the person who developed the system and lead to a loss in business.

Keeping all these things in view, we left no stone unturned in testing our systems. It was tested keeping in view the different possibilities on part of the user. As human beings are prone to commit errors under different working conditions, we had to keep our vigil on different possibilities that can occur on part of the user. The system was tested for validation, functional implementation, and navigation.

Validation Testing

The user must log in to the system with his/her unique login name and password. The user must enter all mandatory fields. If he/she fails to do so then a warning message is issued.

Functional Testing

The entire system was divided into sub-modules. Adding/Updating user Information in the database is done.

Navigational Testing

The system was tested so that all the pages are properly accessible with their respective links.

To uncover the errors in the system we have done testing as follows:

1. Input Checking

In this phase, we tested the validation process only. When users enter the data in the given text box or grids, the proper input format is checked. If entry required numeric data user is bounded to enter only numeric. If text (alphanumeric) data, then the user is bounded to enter text data only also check for null values. Like this, all entries of all input areas are tested.

2. Condition Testing

Condition testing is a method that exercises the logical condition contained in the program module. All relational statements were individually examined and tested. Extreme case values are given for testing.

3. Loop Testing

Loops are cornerstones for the vast majority of all implemented in software. Each loop is examined separately. Its endpoint values were given and the terminating condition each case tested.

4. Output Testing

The First step of testing is checking how friendly it is. Then its accuracy is checked, that is whether it can be present all relevant information, it can report missing less, data, etc.

5. Acceptance Testing

In this type, we run the system live data by the actual user.

5.1 Testing Techniques

Analyze the working of the developed system after implementation is known as testing. There are few techniques we use they include:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing
- Black Box Testing
- White Box Testing
- Grey Box Testing

5.1.1 Unit Testing

It is a type of testing that may occur on the completion of every single module of your system. We also test every single module stand-alone making sure the proper working of the module.

5.1.2 Integration Testing

Integration testing is a type of testing that occurs when two or more modules are combined. This is necessary because if two modules are not working properly together then your system fails. Hence we test multiple probability modules for the proper working of the system.

5.1.3 System Testing

Involves in-house testing of the entire system before delivery to the user. It aims to satisfy the user the system meets all requirements of the client's specifications.

5.1.4 Acceptance Testing

It is pre-delivery testing in which the entire system is tested at the client's site on real-world data to find errors.

5.1.5 Black Box Testing

The technique of testing without having any knowledge of the interior working of the application is black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black-box test a tester will interact with the system user's interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

5.1.6 White Box Testing

White box testing is the detailed investigation of the internal logic and structure of the code. White box testing is also called glass testing or open box testing. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

5.1.7 Grey Box Testing

Grey Box Testing is a technique to test the application with limited knowledge of the internal workings of an application. In software testing, the term more you know the better carries a lot of weight when testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's Customer interface, in Grey box testing, the tester has access to design

documents and the database. Having this knowledge, the tester can better prepare test data and test scenarios when making the test plan.

5.2 Test Approach

Testing can be done in two ways:

5.2.1 Bottom-up Approach

Testing can be performed starting from the smallest and lowest level modules and proceeding one at a time. For each module in bottom-up testing, a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom-level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower-level modules.

5.2.2 Top-down approach

This type of testing starts from upper-level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper-level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower-level module.

5.3 Test Cases

A test case is a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. A test case could simply be a question that you ask of the program. The point of running the test is to gain information, for example, whether the program will pass or fail the test. The test case is the cornerstone of Quality Assurance whereas they are developed to verify the quality and behavior of a product.

TC - 1: Start website and Signup

Table 4 Start Website and Sign Up

Number TC-1	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 01

Description: The requirement is to start the website by Signing up.

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-1	Signup	Users have to Sign up by providing the required information	If information is correct and is incorrect format, the user has to be registered	No Difference, from Expectation	Pass	15-06-2021

TC - 2: Sign Up using incorrect information (Validation Check)

Table 5 Sign Up Using Incorrect Information

Number TC-2	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 02

Description: The requirement is to Sign up using incorrect information to check validation.

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-2	Signup	Users have to Sign up by providing the required information but using incorrect format	If information is correct and is incorrect format, the user has to be registered	No Difference, from Expectation	Pass	15-06-2021

TC - 3: Sending Website feedback

Table 6 Anyone can send feedback about the website

Number TC-3	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 03

Description: The requirement is to Sign up using incorrect information to check validation.

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-3	Contact Us	User has to send an email by providing required information	If information is correct and is incorrect format, the user has to be registered	No Difference, from Expectation	Pass	15-06-2021

TC - 4: Login into Website

Table 7 Login into Website

Number TC-4	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 04
Description: The requirement is to Login using an Email address and password.

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-4	Login	User has to click the Login button and then have to enter email and password to log in	Then a screen appears that shows several functions that a user can perform	No Difference, from Expectation	Pass	15-06-2021

TC - 5: Add to Cart

Table 8 Add to Cart

Number TC-5	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 05

Description: The requirement is to add a book after clicking on the button

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-5	Add to Cart	The user selects a book and clicks add to cart button	Book is added to the shopping cart	No Difference, from Expectation	Pass	15-06-2021

TC - 6: Add to Cart (Without Login)

Table 9 Add to Cart Without Login

Number TC-6

Created Date: 15-06-2021

Created by: Group E-134

Requirement # 06

Description: Requirement is to add a book from guest and then have to signup

Step	Type/name screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-6	Add to Cart	The guest selects a book and clicks add to cart button	The user should create an account.	No Difference, from Expectation	Pass	15-06-2021

TC - 7: Edit Cart

Table 10 Edit Cart

Number TC-7	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 07
Description: Requirement is to change books quantity by a user

Step	Type/nam e	Description	Expected	Actual, if different from Expected	P/F	Date
TC-7	Edit Cart	User changes the Quantity	Quantity and total cost of Cart should be updated	No Difference, from Expectation	Pass	15-06-2021

TC - 8: Upload a new or old Book

Table 11 Upload a new or old Book

Number TC-8

Created Date: 15-06-2021

Created by: Group E-134

Requirement # 08

Description: Requirement is to upload a new or book by Admin

Step	Type/nam e screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-8	Upload a Book Details	Admin can upload details of new or old book	Book should be updated in Categories list	No Difference, from Expectation	Pass	15-06- 2021

TC - 9: Delete a new or old Book

Table 12 Delete a new or old Book

Number TC-9	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 08

Description: Requirement is to delete a new or book by Admin

Step	Type/nam e screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-8	Delete a Book Details	Admin can delete details of new or old book	Book should be deleted from Categories list	No Difference, from Expectation	Pass	15-06- 2021



TC-10: Create a books Category

Table 13 Create a book Category

Number TC-10	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 09

Description: Requirement is to Create a Category by Admin

Step	Type/name screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-9	Create a book Category	Admin adds a new category	Category should be updated to system	No Difference, from Expectation	Pass	15-06-2021

TC - 11: Delete a books Category

Table 14 Delete a book Category

Number TC-11	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 10

Description: Requirement is to delete a Category by Admin

Step	Type/nam e screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-9	Delete a book Category	Admin deletes a new category	Category should be updated to system	No Difference, from Expectation	Pass	15-06- 2021

TC -12: Manage Orders

Table 15 Manage Orders

Number TC-12	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 11
Description: Requirement is to manage orders

Step	Type/nam e screen	Description	Expected	Actual, if different from Expected	P/F	Date
TC-11	Create a book Category	Admin accepts or delete an order	Order is processed according to instructions by Admin	No Difference, from Expectation	Pass	15-06-2021

TC-13: Generate Reports

Table 16 Generate Reports

Number TC-13	Created Date: 15-06-2021
Created by: Group E-134	

Requirement # 12

Description: Requirement is to generate reports or orders and other functions

Step	Type/name	Description	Expected	Actual, if different from Expected	P/F	Date
TC-12	Reports	Reports should be generated by system	Admin can generate reports from system	No Difference, from Expectation	Pass	15-06-2021

5.4 Validation and Verification

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input, corresponding error messages are displayed.

In software project management, software testing, and software engineering, verification and validation (VandV) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle.

Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review.

Verification and validation are not the same things, although they are often confused. Boehm succinctly expressed the difference between

Verification: Are we building the product, right?

Validation: Are we building the right product?

According to the Capability Maturity Model (CMMI-SW v1.1),

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase [IEEE-STD-610].

5.4.1 Software Validation

The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements [IEEE-STD-610].

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

5.4.2 From a testing perspective

- Fault – wrong or missing function in the code.
- Failure – the manifestation of a fault during execution.
- Malfunction – according to its specification the system does not meet its specified functionality.

Chapter 6

6.1 Home Page

USER MANUAL

When we run the Online Book Store Website first home page is displayed. The home page will appear as below. The user and Admin will have different rights.

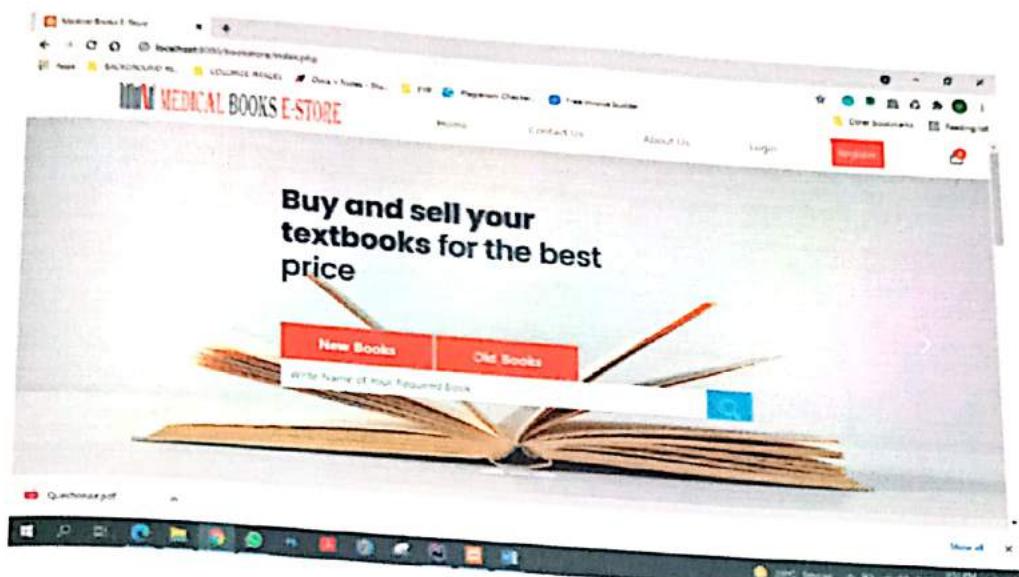


Figure 13 Home Page

6.2 Contact Us Page

When we run the Online Book Store Website Contact Us page is added. If a user faces any kind of issue or has any suggestion, he can send it using the Contact Us page.

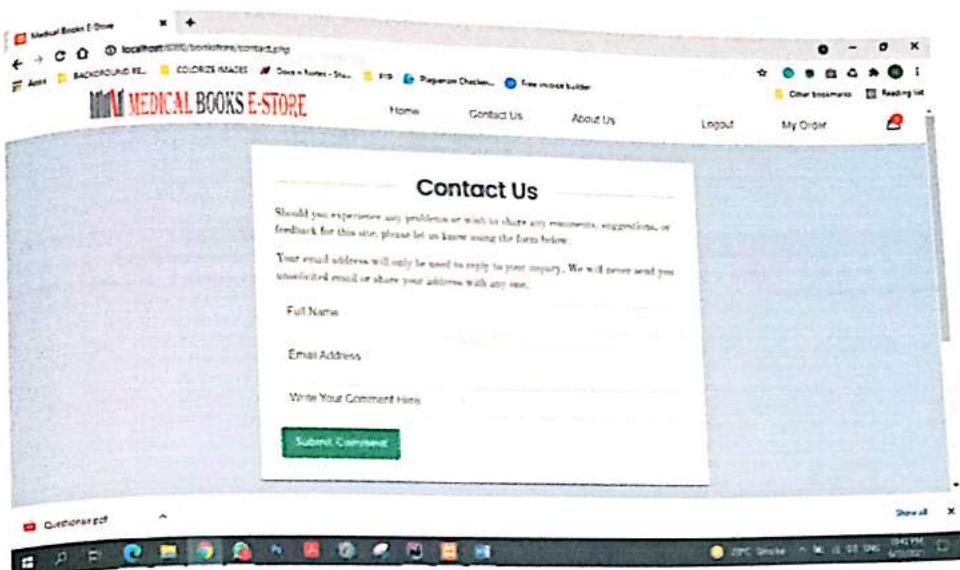


Figure 14 Contact Us

6.3 Register Page

If you are a new user, you can register using the register link or if you are already a user you can log in to purchase a book and pay using the checkout. Any common user can use the search option on the home page to search for a book of his choice by entering a Title of the book.

A user should enter all the required field information. If he didn't fill all the fields he cannot create an account.

The register page will appear as below

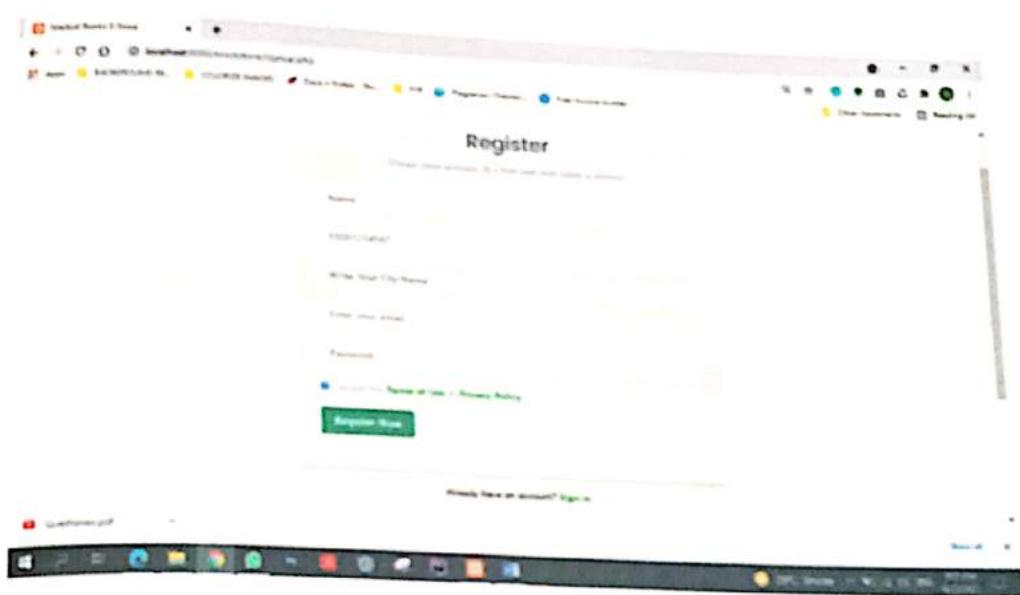


Figure 15 Sign Up Page

6.4 Login Page

After entering all the required data click the submit button to register. Now you can log in to the Website. The login page is used to log in to the website.

The login page will appear as below.

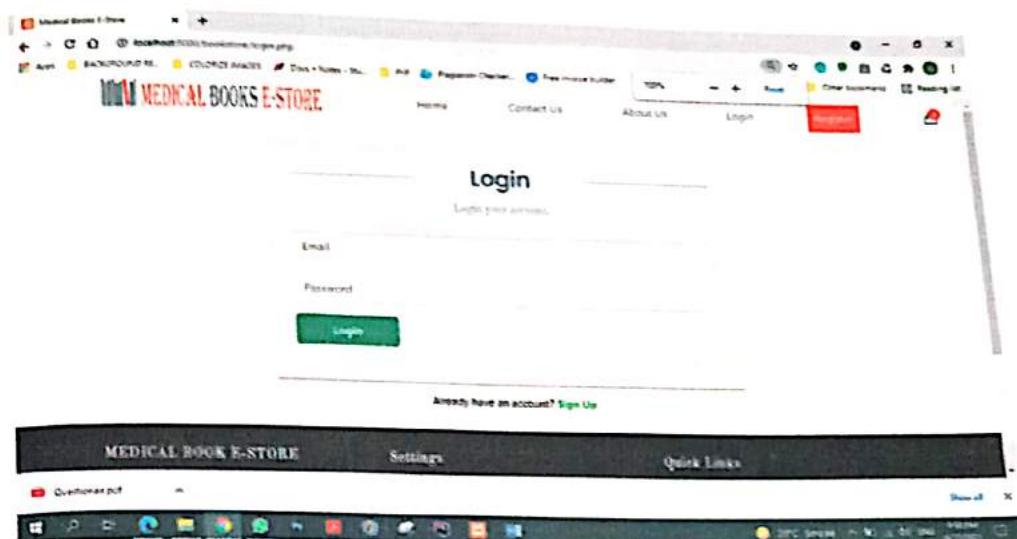


Figure 16 Login Page

6.5 Search a Book

To search for a book, the user can use the search option. By using the search box, a user can search for new books as well as old books.

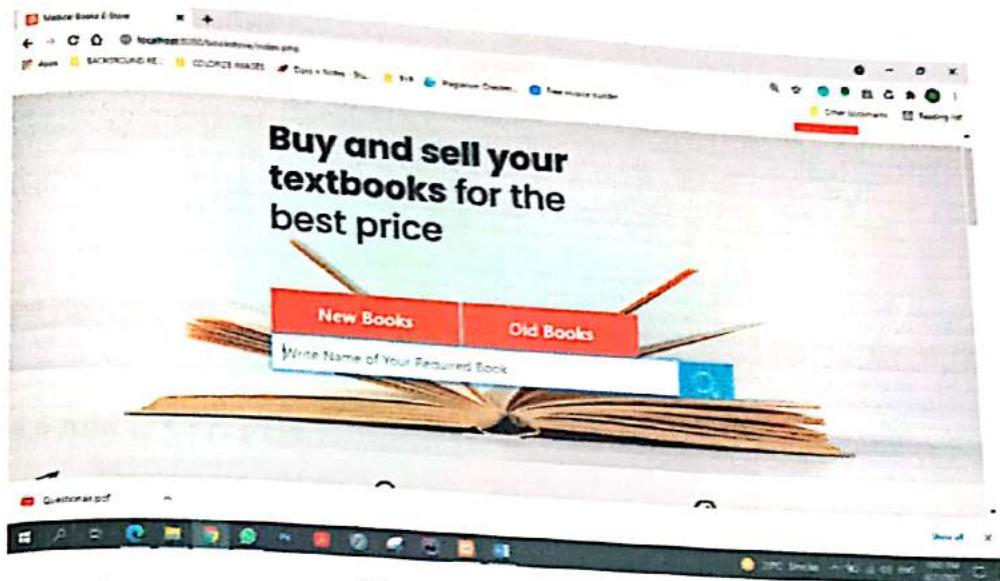


Figure 17 Search a Book

If the user is interested in any particular book he can click the book image to get the book details. The book details page will appear as below.

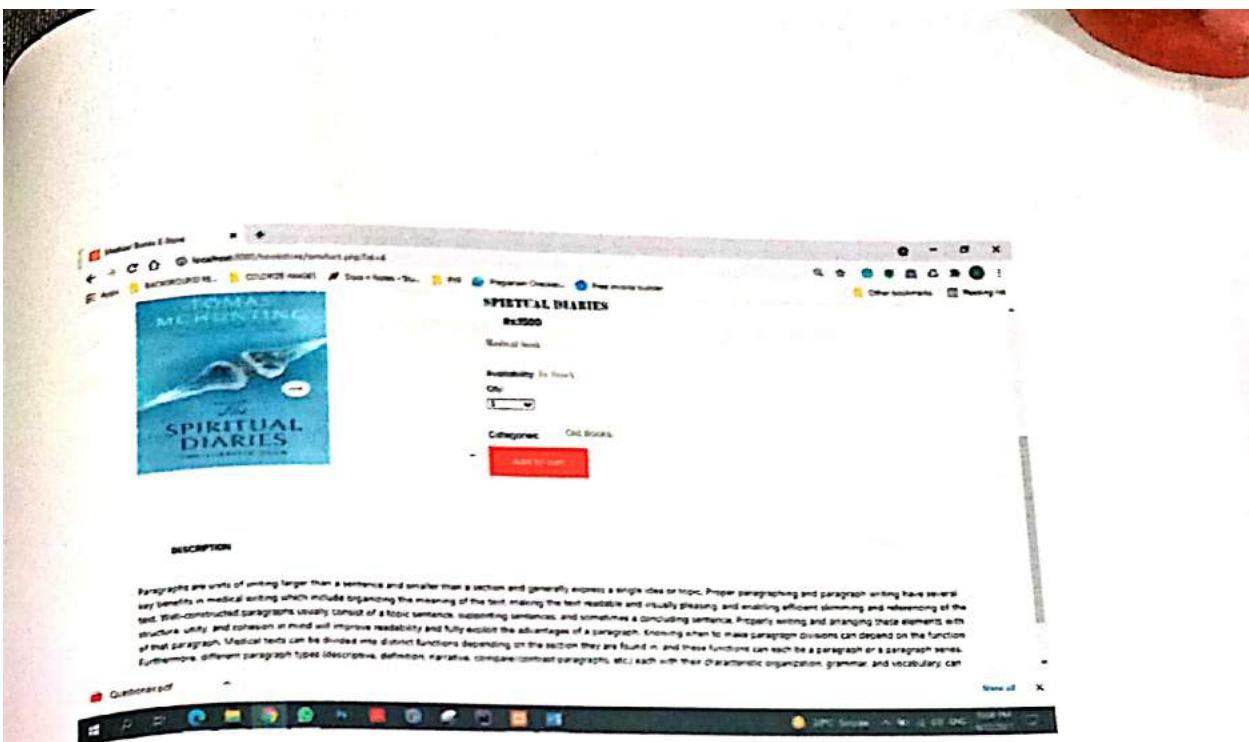


Figure 18 Book Description

6.6 Add to Cart page

The user can add the book to a shopping cart by entering the quantity and clicking the add to shopping cart button. The quantity value should always be positive. The remove button is added so that if a user wants to remove a book user can remove it easily.

PRODUCTS	NAME OF PRODUCTS	PRICE	QUANTITY	TOTAL	REMOVE
	Spiritual Diaries 1500	Rs.1500	1	1500	

Figure 19 Add to Cart

The user can also edit his shopping cart details. If he wants to change the quantity of order user have to click on the Continue Shopping button. When the user clicks the Continue Shopping button on the shopping cart page he will be redirected to the edit shopping cart page.

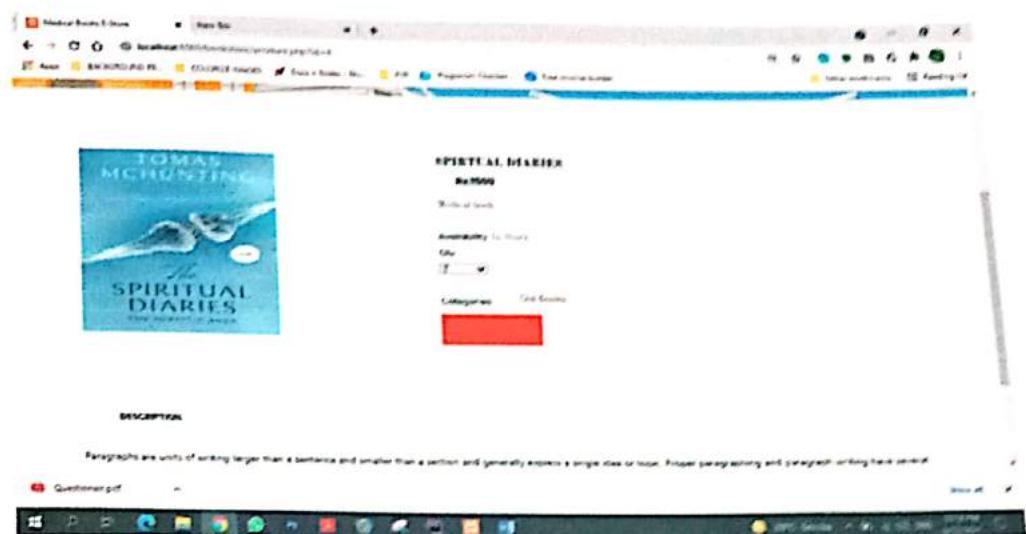


Figure 20 Check Out

6.7 Payment Page

After clicking on the Checkout button payment page will be displayed to a user by using a QR code or other payment option a user will be able to pay his dues.

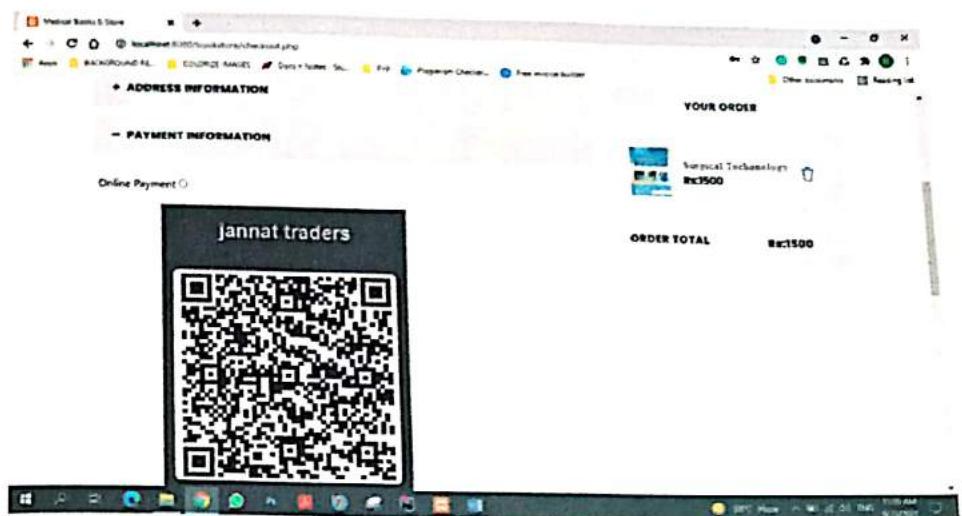


Figure 21 Payment

6.8 Admin Login

The Admin should log in to the Website to use his administrative options. After logging in he will be redirected to the admin menu page.

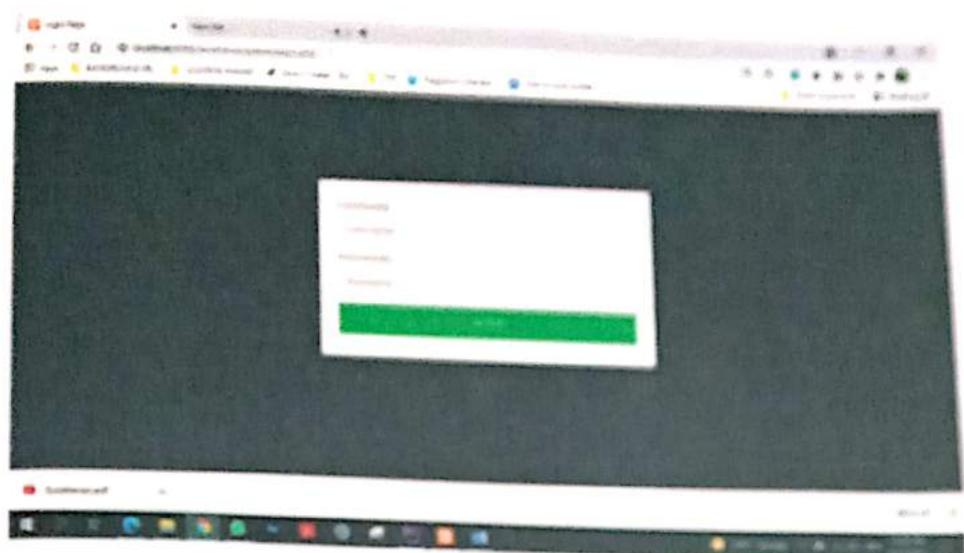


Figure 22 Admin Login Page

6.9 Admin Dashboard

The Administrative options will be displayed in the menu. He can change the details of the books, orders, Categories. And also can process the order and add new books and categories to the books list.

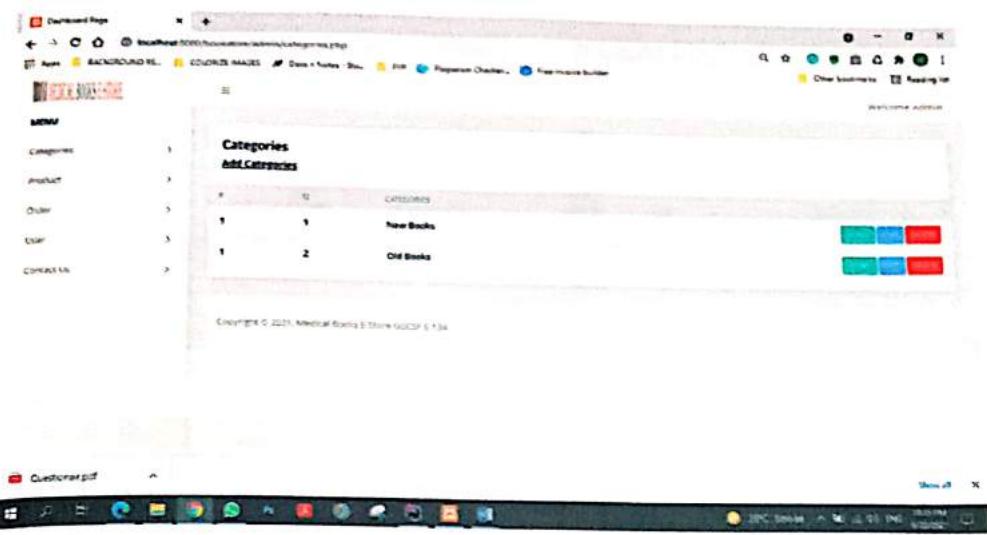


Figure 23 Admin Dashboard

The Admin will be directed to the books list page where he can update, delete and add book details. He can also search for a particular book using this page.

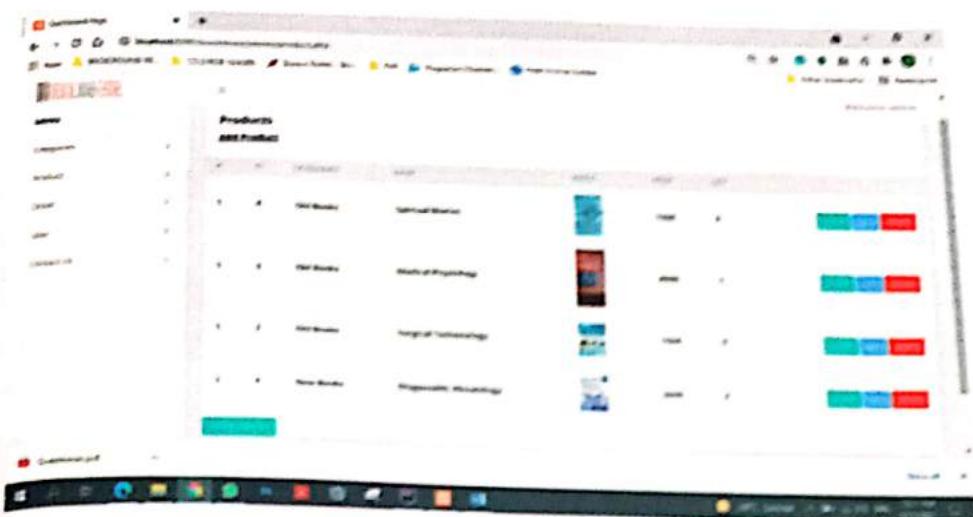


Figure 24 Manage Books