A Project Report on

"ONLINE BOOK STORE"

Submitted in partial fulfillment of requirement for the award of degree

Bachelor of Computer Application

Of

Mangalore University

K Lakshitha Monnappa	Muthamma P K	Sachitha B.S
(186381946)	(186381955)	(186381916)

Internal Guide

Mr.Soorya Narayana P S

Vivekananda degree college

Puttur D.K.

External Guide

Mr.Krishnaraj

AgileTEC

Mangalore

Submitted in the year
2020-21

VIVEKANANDA DEGREE COLLEGE

PUTTUR D.K -574203

https://vcputtur.ac.in

Office No: 4,
First Floor,
Lalbagh Commercial Complex,
Lalbagh , Mangalore-575003
0824-4260555 / 7975263562

CIN: U72200KA2015PTC084828
Ref No: HR/Edu/2021/060

September 25, 2021

To Whomsoever It May Concern:

This is to certify that **Ms. Sachitha B S (Reg No: 186381916), Ms. Muthamma P K (Reg No: 186381955), Ms. K Lakshitha Monnappa (Reg No: 186381946),** final year students of Vivekananda College, Puttur has successfully completed the project work named "**Online Book Store**" at AgileTEC, Mangaluru. We wish them every success in their life and career.

With Thanks and Regards,

AgileTEC Mangaluru Ph : 0824-4260555 A Unit of DevAppSys IT Sol Pvt. Ltd.

Mr. Krishnaraj,

Managing Director,
AgileTEC,
Mangaluru.

DECLARATION

We hereby declare that the project report entitle “ONLINE BOOK STORE” has been proposed by us during the year 2020-2021 under guidance and supervisions of our internal guide Mr Soorya Narayana P S of computer science Department and project guide Mr.Krishnaraj, Managing Director of AgileTEC in partial fulfilment for the award of Bachelor in Computer Applications.

We also declare that this Project is result of our results and has not been submitted to any other university for the award of any Degree or diploma

Place: Puttur — K Lakshitha Monnappa

Date: — Muthamma P K

Sachitha B S

ACKNOWLEDGMENT

When a task is completed we have sought of satisfaction which will be completed if we don't mention the unseen hands that helped us to fulfill our task.

We would like to express our sincere gratitude to our beloved principle Mr.Vishnu Ganapathy Bhat, Vivekananda college for granting us permission to do the project.

We express our gratitude to HOD of Computer Science department Mr.Prakash Kumar for his encouragement and support.

We are so thankful to our project guide Mr Soorya Narayana P S of Computer Science department in helping us to complete the project.

We extend our heartfelt thanks to Mr.Krishnaraj ,Chief Technical developer who gave the idea of developing software.

Finally we would like to express thanks to all our friends who played a major role in regard to completion of this project.

K Lakshitha Monnappa

Muthamma P K

Sachitha B S

INDEX

Online Book Store

CHAPTER-1
INTRODUCTION

1.1 Introduction of the System

1.1.1 Project Title:

"Online Book Store"

1.1.2 Project Category:

Web Application

1.1.3 Overview

In this era of highly advanced software, one can observe a lot of progress in the zone of architectural design and its principles. There are various innovative and efficient software emerging out in the market which have made the life of a common man very simple and easy. Our Online Bookstore is one such software. It is a combination of both e-commerce and book sales industry put together in a single software. Using the online bookstore system has a lot many benefits. There is absolutely no necessity for a consumer to go out looking for a particular book. The book of his/her liking can be easily purchased using the online bookstore software sitting in their comfort zone and just running the software on a system with an active internet connection and a web browser. This helps in saving a whole lot of time and energy of the consumer. Our project is designed and implement using HTML and PHP languages. The database is managed using MySQL and the client-server communication is handled by XAMPP.

1.2 Background

1.2.1 Introduction of the company

Agiltech technologies just bud,but blossoms every day to fulfill the dreams of its clients. It supports every moment for the growth of the organization. In a short time,it has managed to build a great rapport with foreign clients who are full of praise and encouragement and have expressed happiness for good and timely services rendered.

1.2.2 Brief note on existing System

The Internet by far plays a major role in people's life. It has drastically improved the quality of life and the standard of living of so many people. It has widened its branches into many different levels and areas. The e-commerce industry is one such branch which has come into spotlight in the recent years. The online book store system has eased the life of so many book lovers by making it easy for them to purchase books online. It is not always feasible to access a traditional

bookstore, it is limited by its operation time, availability of a particular book, its location and most importantly its capacity and the space required to store numerous books. Such drawbacks have led to the evolution of e-commerce industries related to bookstores. Our project is one such simple e-commerce website which houses various books of different categories for a consumer to purchase online.

1.3 Objective of the System

The objective of our project is to develop a basic ecommerce website for the sales of books over the internet. The website allows an user to search for different books of different categories which are available for purchase. It eases the procedure of purchasing by allowing the user to add the book of his/her liking into a shopping cart.

1.4 Scope of the system

In this application the users can view many books of different category. Users who get registered to this web site can get information about the Books to which they want to purchase for. The users can get the information about the all the stock varities of Books at a single place. Thus prevents the users to visit book stores. Once the User get registered, they can view different shops that has been registered to the website. So that User can view the different books as well as compare price of the books.

Admin has access in all the part of the website .He has privilege to manage users and shopkeepers. Admin has control to display and manage shopkeepers and users of the application.

The shopkeeper who wants to register in this website should pay monthly fee In order to get add books to the application. As there will be completion in publicity regarding selling the books to the users.

1.5 Modules used

Our Proposed system has three main modules:

- Admin
- Shopkeeper
- User

1.5.1 Module description

Admin: Admin should have complete knowledge of the website. Admin has the following control over the website.

- ✓ Login
- ✓ Shopkeeper management

- ✓ User management
- ✓ Logout

Shopkeeper: In this project Multiple shopkeepers make use of the website. Each shopkeeper has the following control over the website.

- ✓ Shopkeeper Registration
- ✓ Login
- ✓ View Registered users
- ✓ Add books
- ✓ Update books
- ✓ View order details
- ✓ View requested books
- ✓ View feedback
- ✓ Logout

User : The user can use application to order books. user has the following control over the website.

- ✓ Registration
- ✓ Login
- ✓ Search books
- ✓ View books
- ✓ Request book
- ✓ Add to cart
- ✓ Order books
- ✓ Give feedback
- ✓ Logout

1.6 Data Structure

Our project consist of following tables :

User

- ❖ User_id
- ❖ Name
- ❖ Email_id
- ❖ Contact_no
- ❖ Password
- ❖ Address

- Is_enabled

Book

- Book_id
- Book_name
- Author_name
- Publisher_name
- Book_price
- isActive
- photo
- description
- shopkeeper_id
- book_copy
- book_catagory

Order

- order_id
- user_id
- book_id
- TotalPrice
- Order_date
- Copies
- Is_paid

Feedback

- Id
- User_id
- Feedback
- Book_id
- Date

Shopkeeper

- Shopkeeper_id
- Shop_name
- Contact_no
- Password
- Address

Category

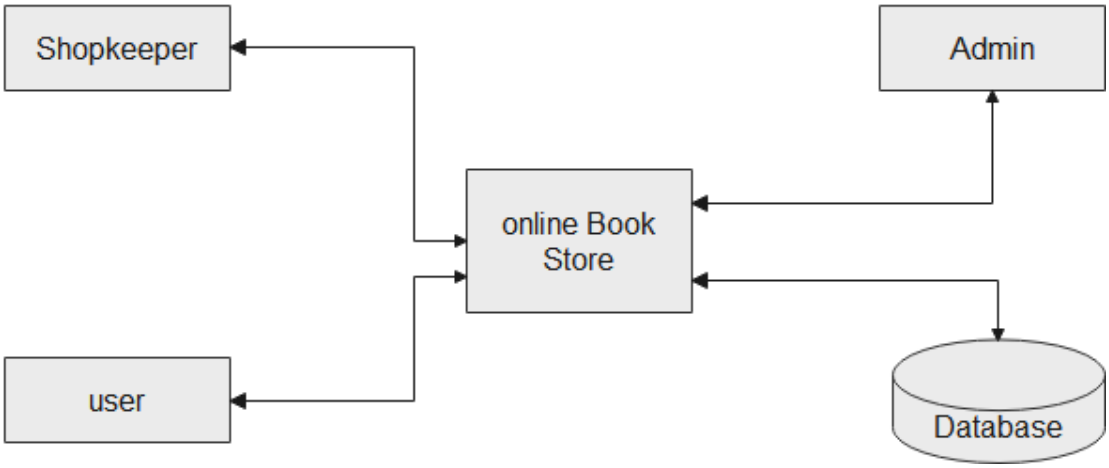
- ❖ Category_id
- ❖ Category_name

Book request

- ❖ book_request_id
- ❖ book_name
- ❖ user_id
- ❖ author_name
- ❖ book_copy

1.7 System Architecture

Architectural design is the process of decomposing a large complex system into small subsystem. These sub systems are meant for providing some related services. The architectural design is basically a layout framework for the system for subsystem control and communication.



1.8 End users

- Shopkeeper
- User
- Admin

1.9 Hardware and Software requirements

1.9.1 Software requirements

- Operating System : Windows 10/ Linux.
- Web Server : XAMPP.
- Programming Languages : PHP, CSS, Java Script.
- Web browsers : Google chrome/Mozilla Firefox/internet explorer.
- Database : MYSQL.

1.9.2 Hardware requirements:

- Processor : 1 GHZ or higher CPU.
- Hard disk : 500 MB available internal storage.
- Memory : 512 MB of RAM is minimally recommended.
- Display : 2.8 inches or larger.

CHAPTER – 2

SOFTWARE REQUIREMENT & SPECIFICATION

2.1 Introduction

SRS stands for Software Requirements Specification, which is a document that fully describes the expected behaviour of a software system. Functional requirements are documented in an SRS, as are non-functional requirements such as performance goals and descriptions of quality attributes. A software requirements specification (SRS) is a description of a software system to be developed. It is modelled after business requirements specification (CONOPS), also known as a stakeholder requirements specification (StRS). The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide to the user for perfect interaction.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

2.2 Overall description

2.2.1 Product perspective:

The system “Online Book Store” is designed to develop a basic ecommerce website for the sales of books over the internet. The website allows a user to search for different books of different categories which area available for purchase. It eases the procedure of purchasing by allowing the user to add the book of his/her liking into a shopping cart. User can buy that book later and do the payment through online payment mode.

2.2.2 Product functionality

The main function of this system is to order books from different shopkeepers. System also supports different payment methods such as COD, UPI etc.

Systems functionalities can be explained as follows:

User Registration

User of the system must register themselves to the system. System takes following data during user registration.

- Name
- Username
- Password
- Mobile Number
- Email Id
- Address

Above registration forms should be validated as below:

- All field should contain values.
- Email ID should be given in correct form.
- Mobile number should be unique and should contain 10 digits.

Login

This is used to login into the system. User and administrator should provide their credentials. It should contain following fields:

- Username
- Password

Form is validated as below:

- Both fields should contain data.
- If credentials are correct home page should be displayed and login page otherwise

Book Module

This module is used to add, delete and update and view book information. Flower add form should contain following fields.

- BookTitle
- Booktype
- Bookdescription
- Cost

- Publisher
- Author

Form is validated as below.

- All fields should contain data.
- Confirmation message should be displayed after adding test data successfully.

User Manager

This module is used to add, delete and update and view user information. View module of this manager should contain following fields.

- Edit
- Delete

Order Manager

This module is used to manage book orders. This module receives user orders and allows admin to accept or reject the order.

2.2.3 User classes and characteristics

Online Book Management System is a web-based application project that helps the book shop owner to manage his customer and business. It includes following end users.

Administrator

The Administrator will handle shopkeeper and user after logging in to the website. That is Admin is allowed to Enable/disable the Shopkeeper or user. Once the user is disabled by the admin,user does not have any access to the system.

Shopkeeper

The shopkeeper will add book related information into the database after logging in to the website. shopkeeper can update the book information based on the modifications in terms of book price and availability. Shopkeeper can also manage user and orders.

User

Users are the customers who orders book for themselves or for others. User need to register the system by providing basic information. After the successful login, user can search book of their choice and add selected book into cart. Finally, they can order the book from their cart and specify the delivery address.

2.2.4 General Constraints

The main constraint here would be the checking the genuineness of the customers, which is not always possible. There can be security risks involved.

- The developed system should run under any platform (Unix, Linux, Mac, Windows etc.) that contains a web browser which supports PHP.
- User must register before he/she login.

2.2.5 Assumptions and Dependencies

- Each user must have an email id and password.
- Each Shopkeeper must have an email id and password.
- Admin must have a Username and password.
- There is only one Administrator.

2.3 Special Requirement

2.3.1 User Interfaces

- Web Application: Admin will use web application to login and communicate with system. System has well defined, user friendly interfaces that allow the admin to operate the system very easily.
- System provides attractive and well-defined interfaces to users.

2.3.2 Hardware Interfaces

- Processor: 1 GHZ or higher CPU.
- Hard disk: 500 MB available internal storage.
- Memory: 512 MB of RAM is minimally recommended.
- Display: 2.8 inches or larger.

2.3.3 Software Interface

- Operating System : Windows 7 or above.
- Web Server : XAMPP.
- Programming Languages : PHP, CSS, Java Script.
- IDE : NetBeans.
- Web browsers : Google chrome/Mozilla Firefox/internet explorer.
- Database : MYSQL.

2.3.4 Communication Interface

Transmission Control Protocol/Internet Protocol is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network (an intranet or an extranet). The entire internet protocol suite -- a set of rules and procedures -- is commonly referred to as TCP/IP, though

others are included in the suite. TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination. TCP/IP requires little central management, and it is designed to make networks reliable, with the ability to recover automatically from the failure of any device on the network.

2.4 Functional Requirements

Proposed system has three main modules :

- **User**
- **Shopkeeper.**
- **Admin**

2.4.1 User Module: User will use the system to search and order book.

- Login Module: This module is used to login into the system .
- Registration module: This module is used to register a new user to the system. System is to be provided with basic information of user during registration.
- Search Module: User uses this module to search a book based on book title or author.
- View Book Module :This module is used to present all available books to user with well-defined interfaces.
- Add to cart : This module is used to add book to the cart.
- Order module: This module is used to order a book from the store.
- Payment Module: This module is used to do payment once he order a book from the store.
- Feedback Module: This module is used by the user to write feedback about purchased book.

2.4.2 Shopkeeper Module: Shopkeeper will use the System to add, update and delete books also to view the details of the user and orders made by users.

- Login Module: This module is used to login into the system.
- Registration module: This module is used to register a new shopkeeper to the system. System is to be provided with basic information of user during registration
- Add Book Module: This allows shopkeeper to add a new book to the system.
- View/Update Module: Shopkeeper uses this module to view and update book information.
- View Order: This module is used by the shopkeeper to view ordered slot and buyer information.
- View Feedback Module: This module is used by the shopkeeper to view user feedback.

2.4.3 Admin Module : Admin will manage shopkeeper and user information.

- Login Module : This module is used to login into the system.
- Shopkeeper module : This module is used to manage all shopkeepers of the system.
- User Module : This module is used to manage all users of the system.

2.5 Design Constraints

- All the inputs should be checked for validation and messages should be givenfor the improper data. The invalid data are to be ignored and error messages should be given.
- Details provided by the user during his sign up should be stored in database.
- While adding the details to the system, mandatory fields must be checked for validation whether the user has filled appropriate data in these mandatory fields. If not, proper error message should be displayed or else the data is to be stored in database for later retrieval.
- User should have the minimum knowledge of accessing the computers.
- User and Administrator should be authenticated by the system.

2.6 Software Quality Attributes

The quality of our website is maintained such a way so that it can be very user friendly to all the users of the website

- Performance: The database used here is robust, reliable & fast. So, users will have to wait for the output very short time.
- Reliability: There is no case of redundancy in the database so it will not take extra memory space.
- Availability: System can be made use of at any time. 24*7 service.
- Maintainability: Maintenance is easy and economical.
- Portability: This system can be run in any operating system and browser.
- Flexibility: The system keeps on updating the data according to the changes that takes place.
- Security: Each time there is a security violation .System restricts the user from accessing that function.
- Timelines: The system carries out all the operations with consumption of very less time.

2.7 Other non-functional requirements

2.7.1 Performance Requirements

- Usability: Software should be usable, without any effort and it should have appropriate user interface.
- Response Time: Software should response within the estimated time to display results to the user.9-
- Software development life cycle: Here agile method is used which combines the advantages of waterfall approach and iterative model.

2.7.2 Safty Requirements

- People other than the registered user should not be able access the system.
- Only the respective user can change their details.
- Authentication is done during the login.
- Whenever user has to edit his profile the UI prompts to enter the user password.
- The system searches for the email id and mobile number during add user. If user Already Exists With Same Email Id And Mobile Number, Then A Message Is Shown.

CHAPTER – 3

SYSTEM DESIGN

3.1 Introduction

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. In System design focus is on deciding which modules are needed for the system, the specifications of these modules should be interconnected is called "System Design".

System design is also called top-level design. Here we consider a system to be set of components with clearly defined behaviour that interact with each other in a fixed manner to produce some behaviour. In a system design, the design consists of module definitions, with each module supporting a functional abstraction.

If the broader topic of product development" blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to the manufactured. Systems design is therefore the process of defining and developing a system to satisfy specified requirements of the user.

3.2 Assumptions and Constraints

3.2.1 Assumptions

The system should be able to install the files in the SQL server. This desktop pplication is installed in the windows and doesn't cost. This project is aimed to provide satisfactory result to the user by providing the required information. Administrator is created in the system manually. Admin alone can make use of the application, providing a valid login id and password.

- The code should be free with compilation errors/syntax errors.
- The details must have an interface which is simple enough to understand.

3.2.2 Constraints

- This system is provisioned to be built on the PHP language which is highly flexible.
- Decision regarding which database to use should be taken considering the fact that data being exchanged or stored is large, and the appropriate data management system will yield efficient performance.

- More space is required to keep all the records.Database should not be overloaded.
- The tables of the database are designed as normalized table.
- Database is not shared.

3.3 Functional decomposition

Functional decomposition refers broadly to the process of resolving a functional relationship into its constituent. Parts in such a way that the original function can be reconstructed (i.e., recomposed) from those parts by function composition. In general, this process of decomposition is undertaken either for the purpose of gaining insight into the identity of the constituent components (which may reflect individual physical processes of interest, for example) or for the purpose of obtaining a compressed representation of the global function, a task which is feasible only when the constituent processes possess a certain level of modularity.

3.3.1 System Software Architecture

The software architecture of a software system is the set of structures needed to reason about the system, which comprise software components, relations among them, and properties of both.

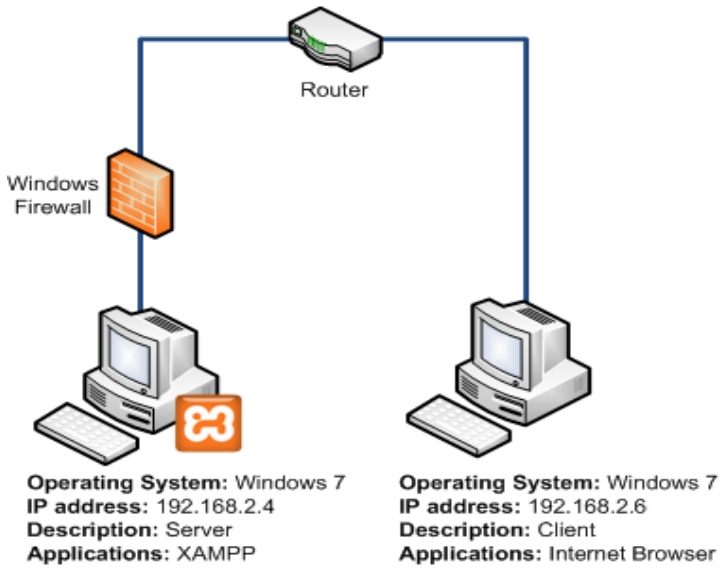


Figure 3.3.1 System Software Architecture

3.3.2 System Technical Architecture

The Apache server used to execute PHP programming and MYSQL server used to store database records. The project “online book store” works under following technical architecture:

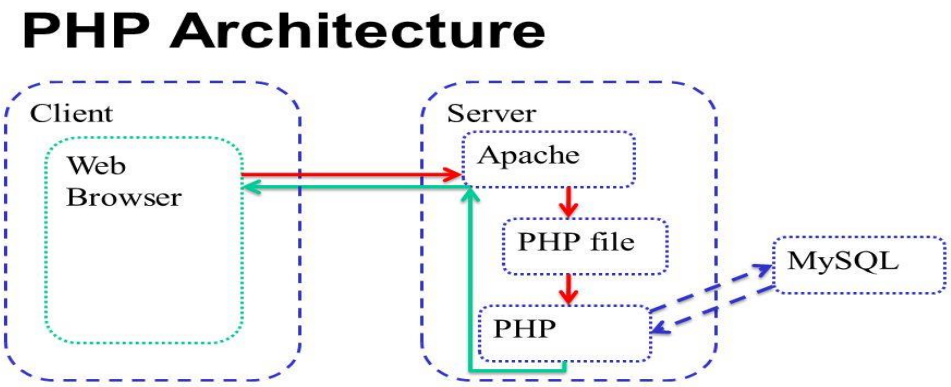


Figure 3.3.2 System Technical Architecture

3.3.3 System Hardware Architecture

Hardware architecture refers to the identification of a system's physical components and their interrelationships. This description often called a hardware design module, allow hardware designers to understand how their components fit into system architecture and provide software component designer important information needed for software development and integration .

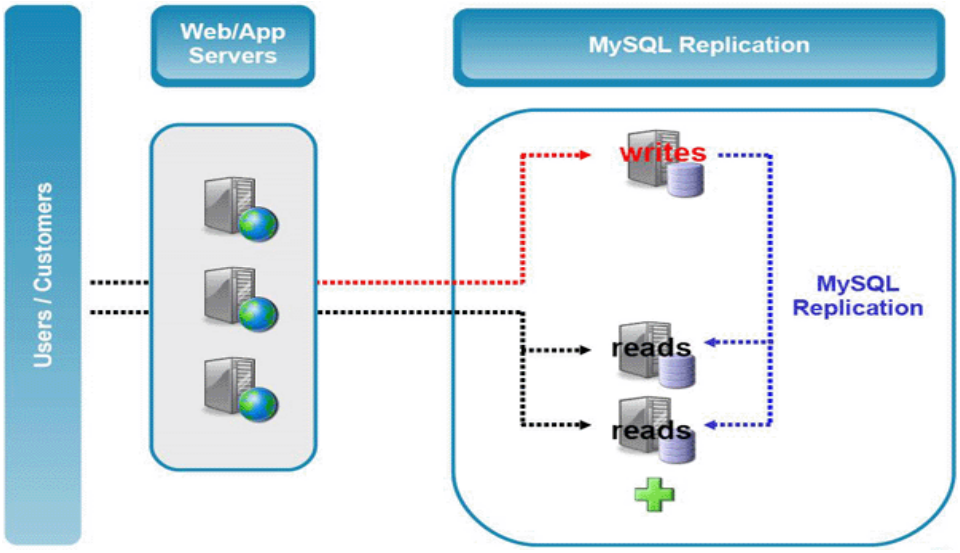


Figure 3.3.3 System Hardware Architecture

3.3.4 External Interfaces

- Name of the application: Online Book Store.
- Details of interface: Admin interface, Shopkeeper interface, User interface.
- Type of Interface: GUI, Menu driven and form based.

3.4 Description of Programs

3.4.1 Context Flow Diagram(CFD)

A context flow diagram (CFD) is a graphical representation of the "flow" of context through an information system.

CFDs can also be used for the visualization of data processing (structured design).On a CFD, data items flow from an external data source or an internal data source to an internal data store or an external data sink, via an internal process. A CFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the control through an algorithm, allowing a reader to determine what operations will be performed, in that order, and under what circumstances.

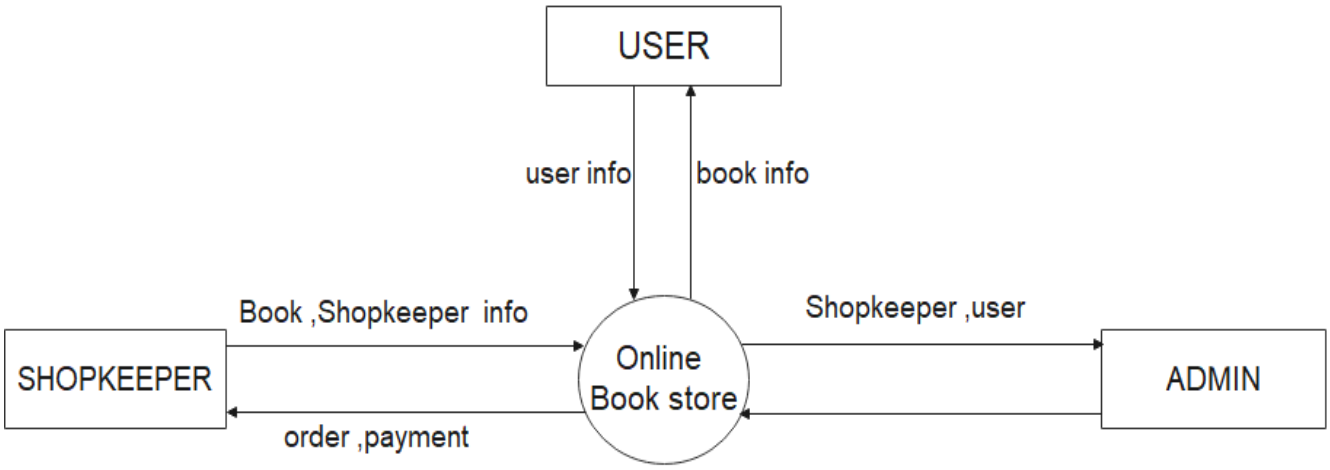


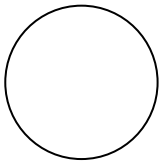
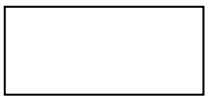
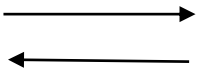
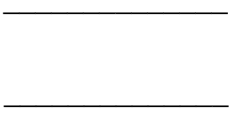
Figure:3.4.1 Context Flow Diagram

3.4.2 Data Flow Diagram

Data Flow Diagram is a graphical representation of a system or a portion of the system.It consists of data flows , process, source and sink and store all the description through the use of easily understandable symbols.DFD is one of the most important modeling tool. It s used to model the system , component that interact with the system , uses the data and information flows in the system. DFD shows the information through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output. DFD is also known as bubble chart or Data Flow Graphs. DFD may be used to represent the system at any level of abstraction. DFD's may partition into a level that represent increasing information flows and functional details.

Rules Regarding DFD Construction:

- A process cannot have only outputs.
- A process cannot have only inputs.
- The input to a process must be sufficient to produce the outputs from the process.
- All data stores must be connected to at least one process.
- All data stores must be connected to a source or sink.
- A dataflow can have only one direction of flow. Multiple data flows to and/or from the same process and data store must be shown by separate arrows.
- If the exact same data flows to two separate arrows, it should be represented by a forked arrow.
- Data cannot flow directly back into the process it has just left. All data must be named using noun phase.

Names	Notations	Discription
Process		A process shows a transformation or manipulation of data flows within the system. A process transforms incoming data flow into outgoing data flow.
External Entity		External entities are outside the system, but they either supply input data into the system or use system output. External entities are represented by a rectangle.
Data Flows		A data flow shows flow of information from source to destination. A data flow is represented by a line, with arrowhead showing the direction of flow.
Data Store		A data store indicates to which database a particular data is stored and from which database the data is retrieved.

a) Level 1 DFD : Admin

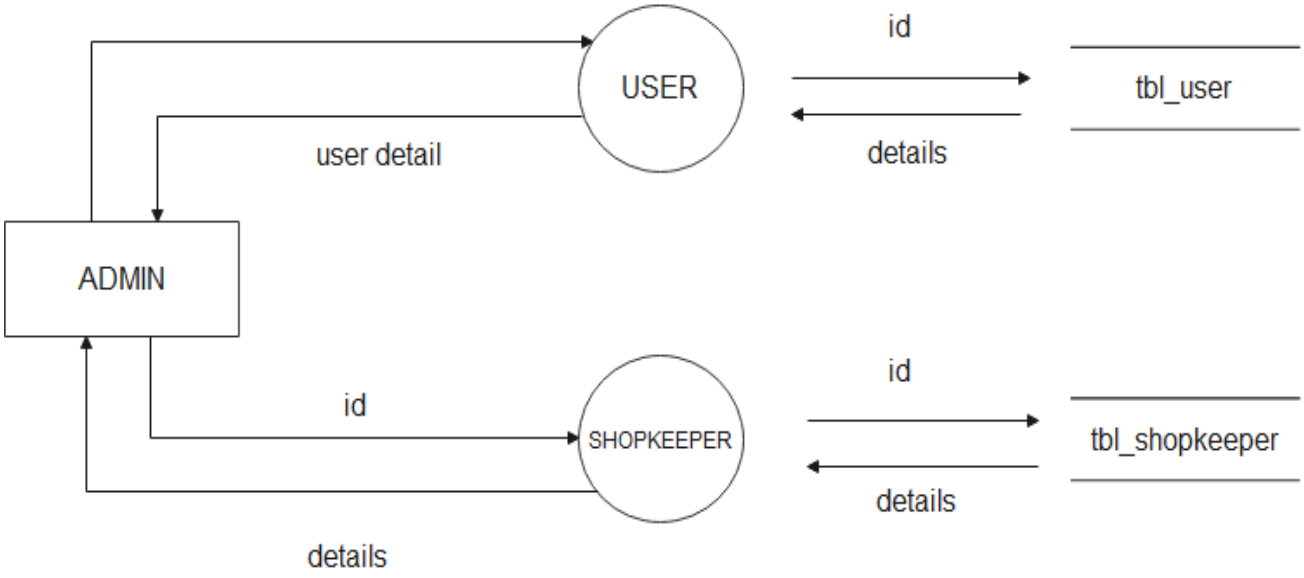


Figure: 3.4.2.1 Data Flow Diagram level 1 for Admin

b) Level 1 DFD : Shopkeeper

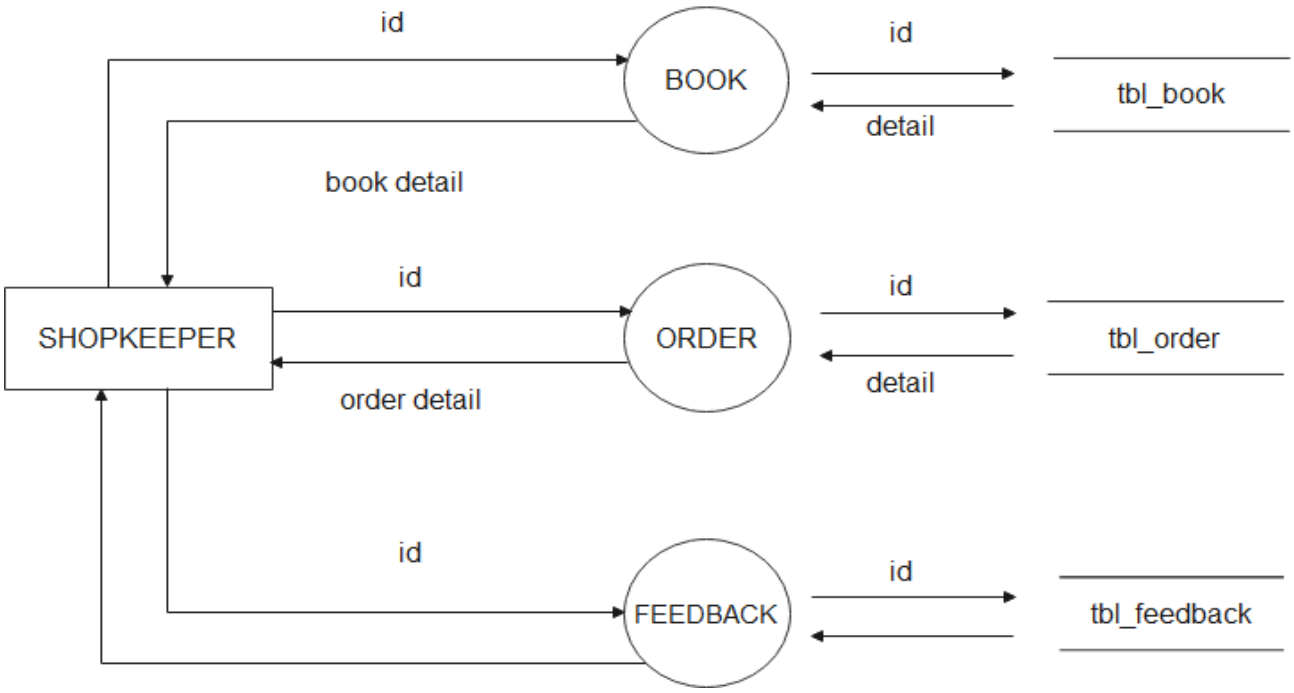


Figure: 3.4.2.2 Data Flow Diagram level 1 for Shopkeeper

c) Level 1 DFD : User

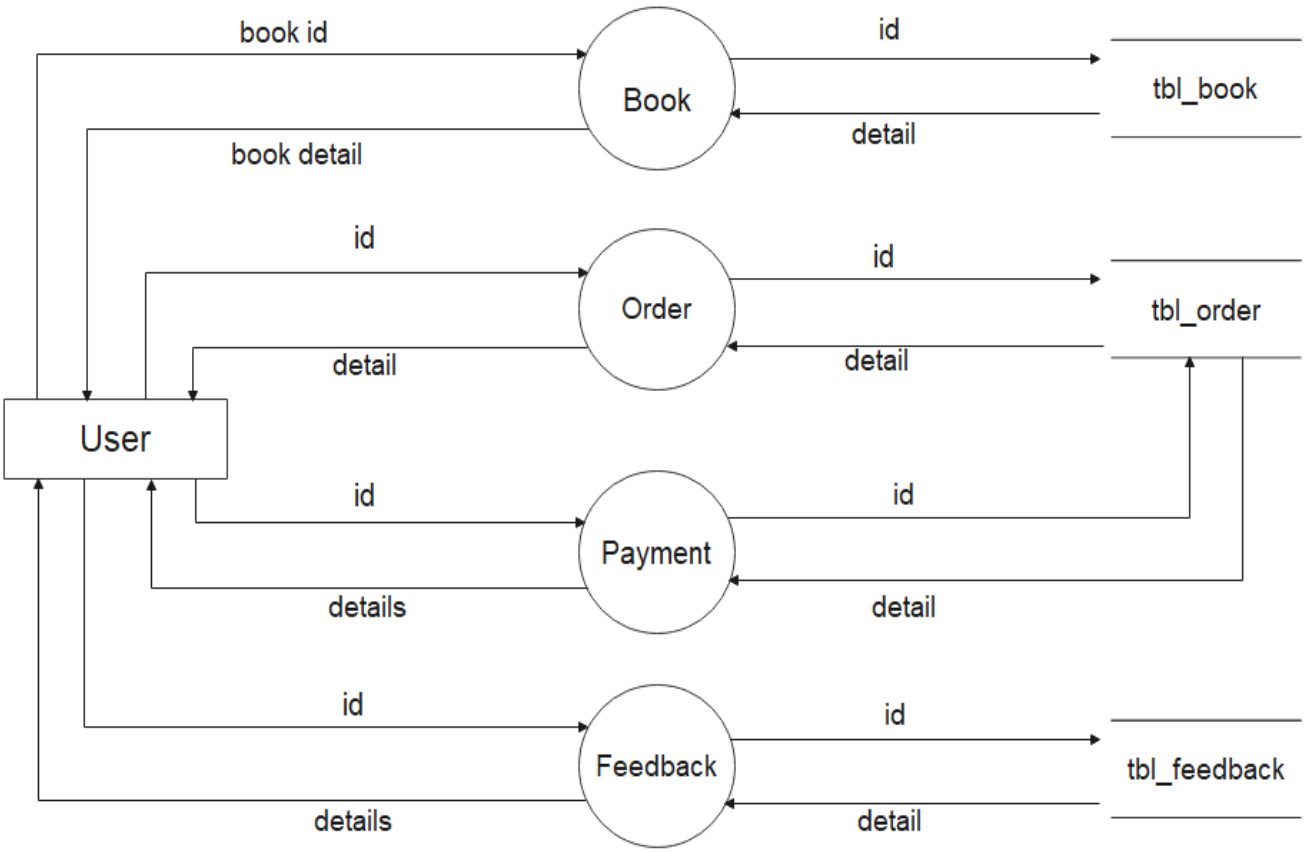


Figure: 3.4.2.3 Data Flow Diagram level 1 for User

d) Level 2 DFD :Admin

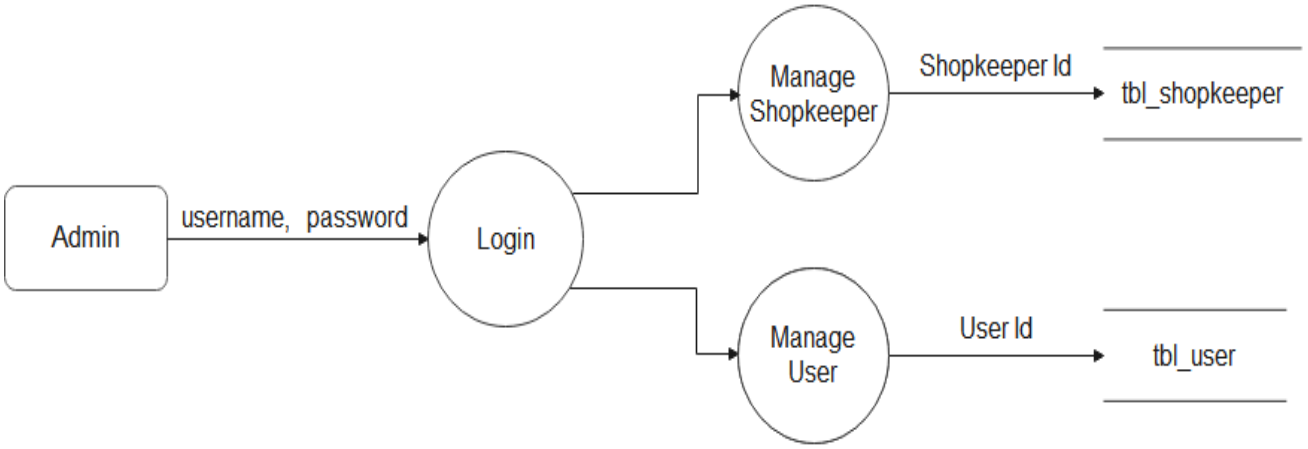


Figure: 3.4.2.4 Data Flow Diagram level 2 for Admin

e) Level 2 DFD : Shopkeeper

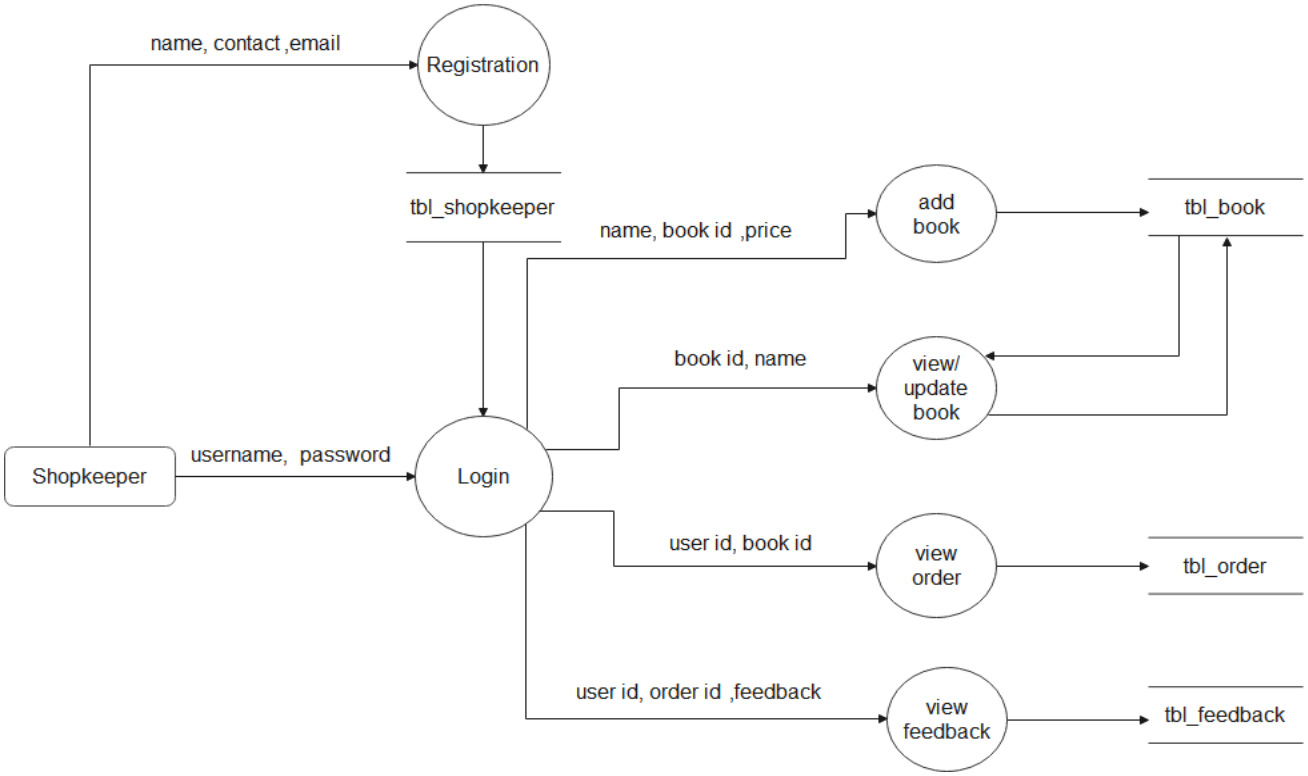


Figure: 3.4.2.5 Data Flow Diagram level 2 for Shopkeeper

f) Level 2 DFD : User

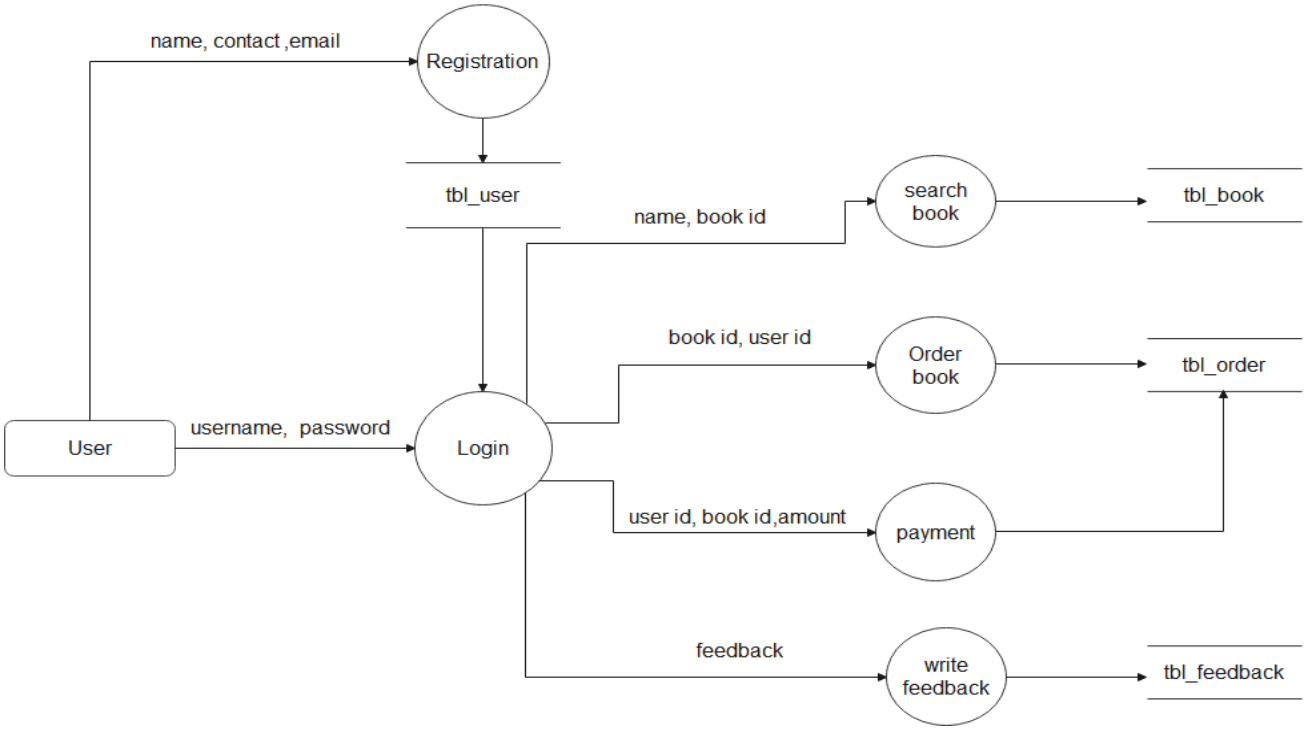


Figure: 3.4.2.6 Data Flow Diagram level 2 for User

3.5 Description of components

- **Functional component 1: Login**

  Input : Username, password.

  Process : This component is to enable the logging in of the admin .The system checks for the correctness of the username and password.

  Output : Admin will be taken to the admin page on successful login.

- **Functional component 2: Shopkeeper**

  Input : Shop Name , Contact No, Email id, password , Address.

  Process : The system validates all the required inputs and enables signing up for the Shopkeeper.

  Output : Shopkeeper will be taken to the home page on successful sign up.

- **Functional component 3: user**

  Input : Name , Contact No, Email id, password , Address.

  Process : The system validates all the required inputs and enables signing up for the User.

  Output : User will be taken to the home page on successful sign up.

- **Functional component 4: Add Book**

  Inpu : Book Name, Author Name, Publisher, Choose category , description, book price,no of copies, choosing the file.

  Process : The system will add book to the application.

  Output : The book added by the shopkeeper is displayed to the user.

- **Functional component 5: order Book**

  Input : No of book copies ,payment type.

  Process : The user oders the desired books.The data is stored in order table.

  Output : The book orders are displayed to the shopkeeper.

- **Functional component 6: Write feedback**

  Input : Choose book, feedback.

  Process : The user writes feedback regarding books purchased .The data is stored in the Feedback table.

  Output : The user feedback is displayed to the shopkeeper.

CHAPTER-4

DATABASE DESIGN

4.1 Introduction

Database design is the process of producing a detailed data model of database. The data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language which can then be used to create a database. A fully attributes data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system principally and most correctly. It can be thought of as the logical design of the database structures used to store the data. In the relational model, these are the tables and views. In an object database, the entities and relationship map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the data base structures, but also the forms and quires used as a part of the overall database application within the Data Base Management System (DBMS).

4.2 Purpose and Scope

The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

The logical model; concentrates on the data requirements and data to be stored independent of physical considerations. It does not concern itself with how the data will be stored physically. The physical data design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as DBMS.

4.3 Database Identification

- Database table name and column names defined without leaving space.
- Lower case used to create database tables name columns names.
- Primary Key and foreign Key defined with same name.

4.4 Schema Information

A Schema is a pictorial represents of the relationship between the databases tables in the database that is created. The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term “schema” refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called Integrity constraints imposed on a

database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language.

Various datatype

S.NO	DATATYPE	DESCRIPTION
1	Int	Representation of integer value that can be stored in 4 Bytes. The range of values is -2,147,483,648 to -2,147,483,647,'int'is the short form of Integer.
2	Varchar	Store variable length Strings of Unicode Character. Each 'Varchar' Character uses 1 byte of storage space.
3	Text	Stores Character String value of maximum 2,147,483,647.
4	Blob	BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files.
5	Date/Time	The 'Date/Time' datatype specifies the date and time, with each value being stored as an integer value in 4 bytes.

4.5 Table Definition

Table Name: User

User_id	name	Email_id	Contact_no	password	address	Is_enabled

Table Name: Book

Book_id	Book_name	Author_name	Publisher_name	Book_price	Is_active	photo

description	Shopkeeper_id	Book_copy	Book_category

Table Name: Order

Order_id	User_id	Book_id	Total_price	Order_date	copies	Is_paid

Table Name: Shopkeeper

Shopkeeper_id	Shop_name	Contact_no	Email_id	password	address

Table Name: Category

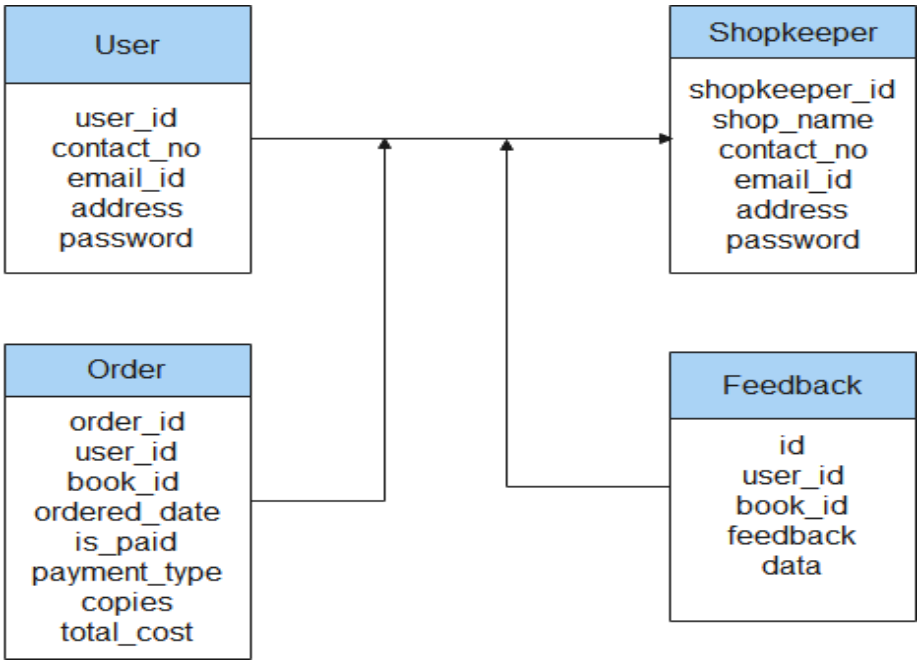
Category_id	Category_name

Table Name: Book request

Book_request_id	Book_name	User_id	Author_name	Book_copy

4.6 Physical Design

It is the process of transforming a logical data model into a physical model of a database.



4.7 Data Dictionary

Table Structure

The database “Online Book Store” is organized into following tables:

- User
- Book
- Order
- Feedback
- Shopkeeper
- Category
- Book request

Table name :User

S.No	Field Name	Data Type	Constraint Type	Desciption
1	UserID	Integer	Primary Key	Unique id of user
2	Name	Varchar2	Not Null	User name

3	Email_id	Varchar2	Not Null	Valid email Address
4	Contact_No	BIGINT	Not Null	Valid number
5	Password	Varchar2	Not Null	password
6	Address	Varchar2	Not Null	Address of the user
7	Is_enabled	Int	Not Null	User has been enabled

Table name : Book

S.No	Field Name	Data Type	Constraint Type	Description
1	Book_ID	Integer	Primary Key	Unique id of Book
2	Book_Name	Varchar	Not Null	Name of Book
3	Author_name	Varchar	Not Null	Name of the Author
4	publisher_name	Varchar	Not Null	Name of Publisher
5	Book_Price	Integer	Not Null	Cost of Book
6	IsActive	Int	Not Null	Indicates Whether the the book is available
7	Photo	Blob	Not Null	Image of the Book
8	Description	Varchar	Not Null	Description of Book
9	Shopkeeper_id	Integer	Not Null	Id of Shopkeeper
10	Book_copy	Integer	Not Null	No of Book copies
11	Book_catagory	Int	Not Null	Catagory of Book

Table name :Order

S.No	Field Name	Data Type	Constraint Type	Description
1	OrderID	Integer	Primary Key	Unique id of Order
2	UserID	Integer	Not Null	Id of user
3	bookID	Integer	Not Null	Id of Book

4	TotalPrice	Integer	Not Null	Total price of the Book
5	Order_date	Date	Not Null	Date on which the book was ordered
6	Copies	Integer	Not Null	No of Book copies ordered
7	Is_paid	Integer	Not Null	Determines whether the payment is done

Table name :Feedback

S.No	Field Name	Data Type	Constraint Type	Description
1	ID	Integer	Primary Key	Unique id of feedback
2	User_ID	Integer	Not Null	Id of user
3	Feedback	Varchar	Not Null	Id of feedback
4	Book_id	Integre	Not Null	Id of Book
5	Date	Date	Not Null	Date on which the feedback was written

Table name : Shop Keeper

S.No	Field Name	Data Type	Constraint Type	Description
1	shop_keeper_rid	Integer	Primary Key	Unique id of shopkeeper
2	Shop_name	Varchar	Not Null	Name of the Book store
3	Contact_no	Varchar	Not Null	Valid mobile umber
4	Email_id	Varchar	Not Null	Valid email address
5	password	Varchar	Not Null	password
6	address	Varchar	Not Null	Book Shop Address

Tabel name : Catagory

S.No	Field Name	Data Type	Constraint Type	Description
1	Category_ID	Integer	Primary Key	Unique id of catagory
2	catagory_name	Varchar2	Not Null	Category name

Table name : Book request

S.No	Field Name	Data Type	Constraint Type	Decription
1	book_request_rid	Integer	Primary Key	Unique id of the Requested Book
2	Book_Name	Varchar	Not Null	Requested Book Name
3	User_id	Integer	Not Null	Id of the user
4	author_name	Varchar	Not Null	Book author name
5	Book_copy	Integer	Not Null	No of Book copies

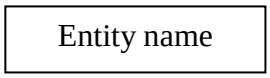
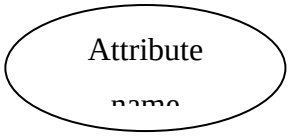
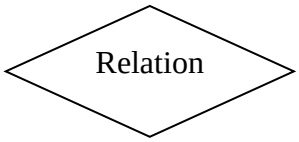
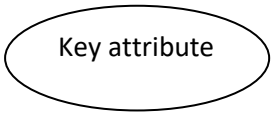
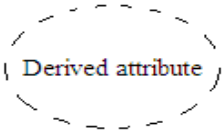
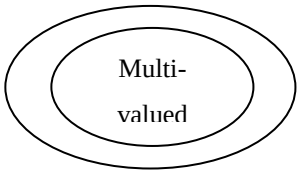
4.8 Entity-Relationship Diagram:

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

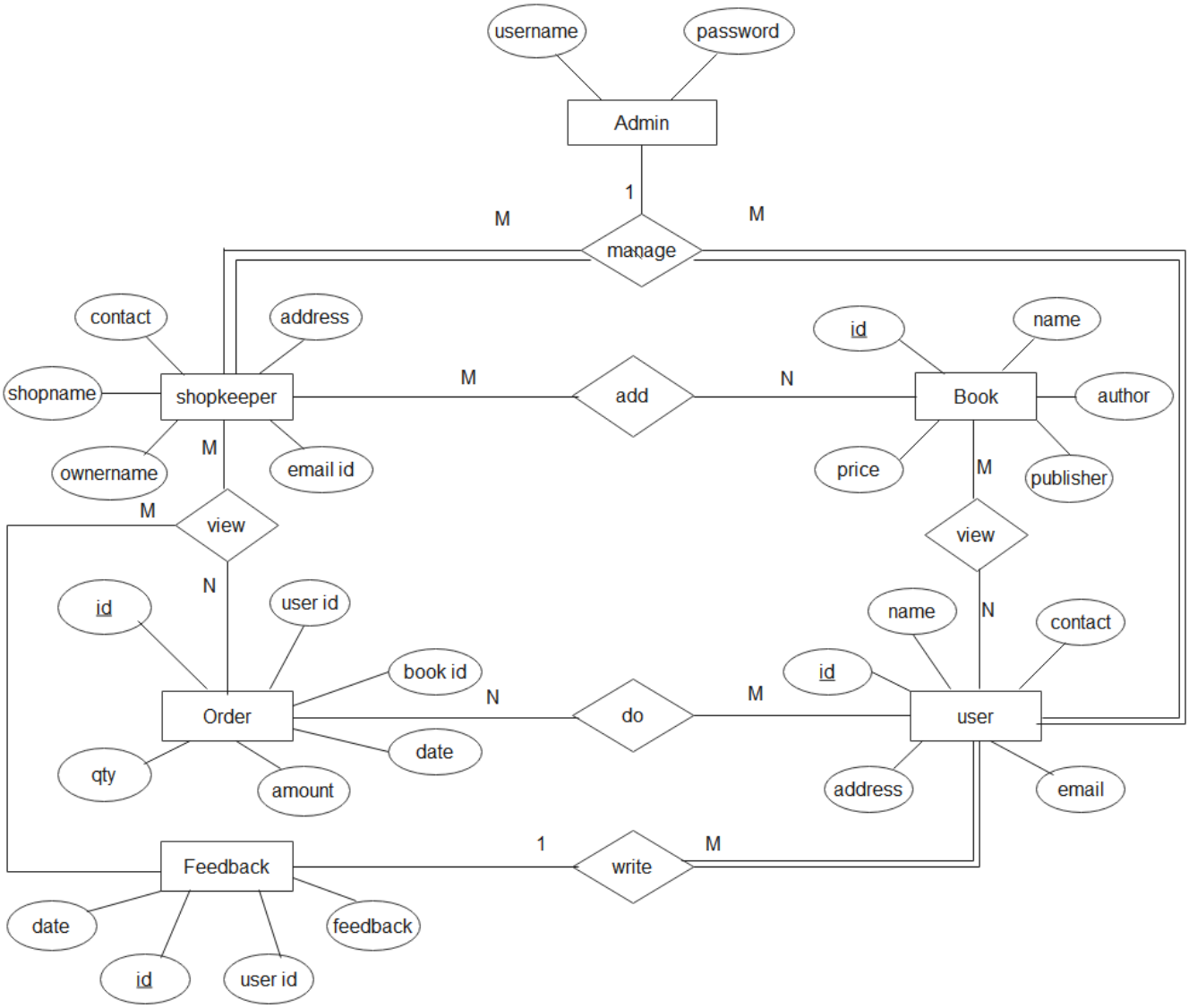
ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure

ER-Diagram Symbols

Name	Symbols	Discription
Entity		It may be an object with the physical existence or conceptual existence. It is represented by a Rectangle.
Attribute		The properties of the entity can be a attribute. It is represented by a Ellipse.
Relationship		Whenever an attribute of one entity refers to another entity, some relationship exists. It is represented by a Diamond.
Key attribute		An entity type usually has an attribute whose values are distinct for each individual entry in the entity set. It is represented by a Underlined word in ellipse..
Derived Attribute		Dashed ellipse denotes derived attributes.
Multi valued Attribute		An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in an ER Diagram
Link		Lines link attribute to entity sets to relation
Cardinality Ratio	1:1 1:M M:1 M:M	It specifies the maximum number of relationships instances that an entity can participate in. There are four cardinality ratios.

ER Diagram



4.9 Database Administration

4.9.1 System Information

- Server : localhost via TCP/IP.
- Server type : MySQL.
- Server version : 5.6.12-log - MySQL Community Server (GPL).
- Protocol version 10.
- User : root@localhost.
- Server charset : UTF-8 Unicode (utf8) .

4.9.2 Database Management System (DBMS) Configuration

The exact set of database administration duties of each DBA is dependent on his/her job profile, the IT policies applied by the company he/she works for and last but not least-the concrete parameters of the database management system in use. A DBA must be able to think logically to solve all problems and to easily work in a team with both DBA colleagues and staff with no computer training.

XAMPP Version: 3.2.1

MySQL version: 5.6.12

4.9.3 Support Software

The system installs MYSQL server while installing XAMPP software. All the backup content stores in MYSQL data folder.

4.9.4 Storage Requirements

- Storage for databases constitutes a set of compatible software and hardware where database files are stored.
- Compatibility must be carefully tested to ensure bottlenecks and the possibility of data corruption.
- Failure free work in high load conditions and the redundancy of vulnerable components must be provided. Such a set must meet the highest standard in terms of performance and reliability, to ensure continuous and fast access to import data.

4.9.5 Backup and Recovery

A volatile storage like RAM stores all the active logs, disk buffers, and related data. In addition, it stores all the transactions that being currently executed. If such a volatile storage crashes abruptly, it would take away all the logs and copies of database. It makes recovery almost impossible, as everything that is required to recover the data is lost.

Following techniques may be adopted in case of volatile storage-s

- We can have checkpoints at multiple stages so as to save the database contents periodically.
- A state of active database in the volatile memory can be periodically dumped onto a stable storage, which may also contain logs and active transactions and buffer blocks.
- When a system recovers from a failure, it can restore the latest dump.
- It can recover the system by consulting undo-list to restore the state of all transactions up to the last checkpoint

CHAPTER-5

DETAILED DESIGN

5.1 Introduction

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us towards how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases, particularly testing and maintenance.

The design activity often results in three separate outputs- Architecture Design, High Level Design and Detailed Design. Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired results. The High Level Design identifies the modules that should be built for developing the system and the specifications of these modules. At the end of system design all the major data structures, file formats, output formats etc., are also fixed. In detailed design, the internal logic of each of the modules are specified.

The design process for the software systems often has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules and how the modules should be interconnected. This is what is called the System Design or the Top-Level Design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied is decided. This design level is often called Detailed Design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

The major tasks identified for carrying out the detailed design activity include:

1. Understanding the architecture and requirements
2. Creating detailed designs
3. Evaluating detailed designs
4. Documenting software design
5. Monitoring and controlling implementation

5.2 Structure of the Software Package

The functional components identified during the system design are listed here:

- Functional Component 1 :Admin
- Functional Component 2 :Shopkeeper
- Functional Component 3 :User

a) Admin module

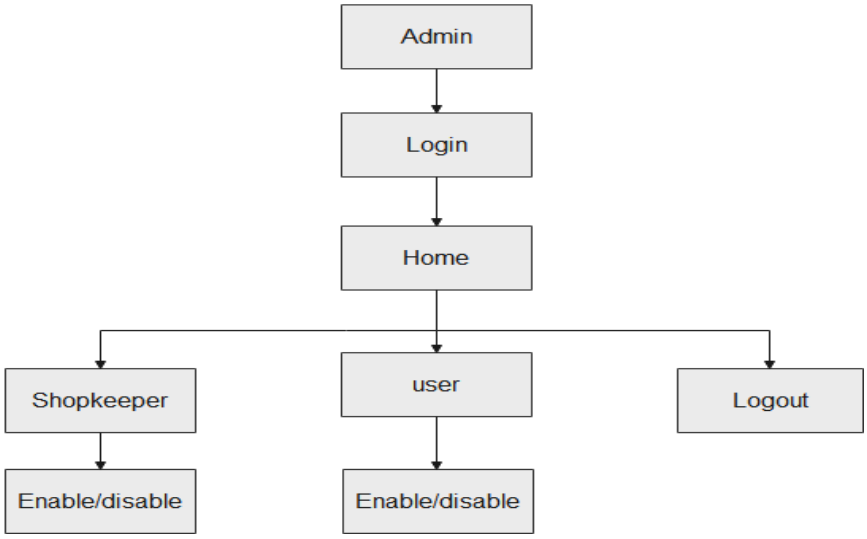


Figure 5.2.1 Structure chart for Admin

b) Shopkeeper Module

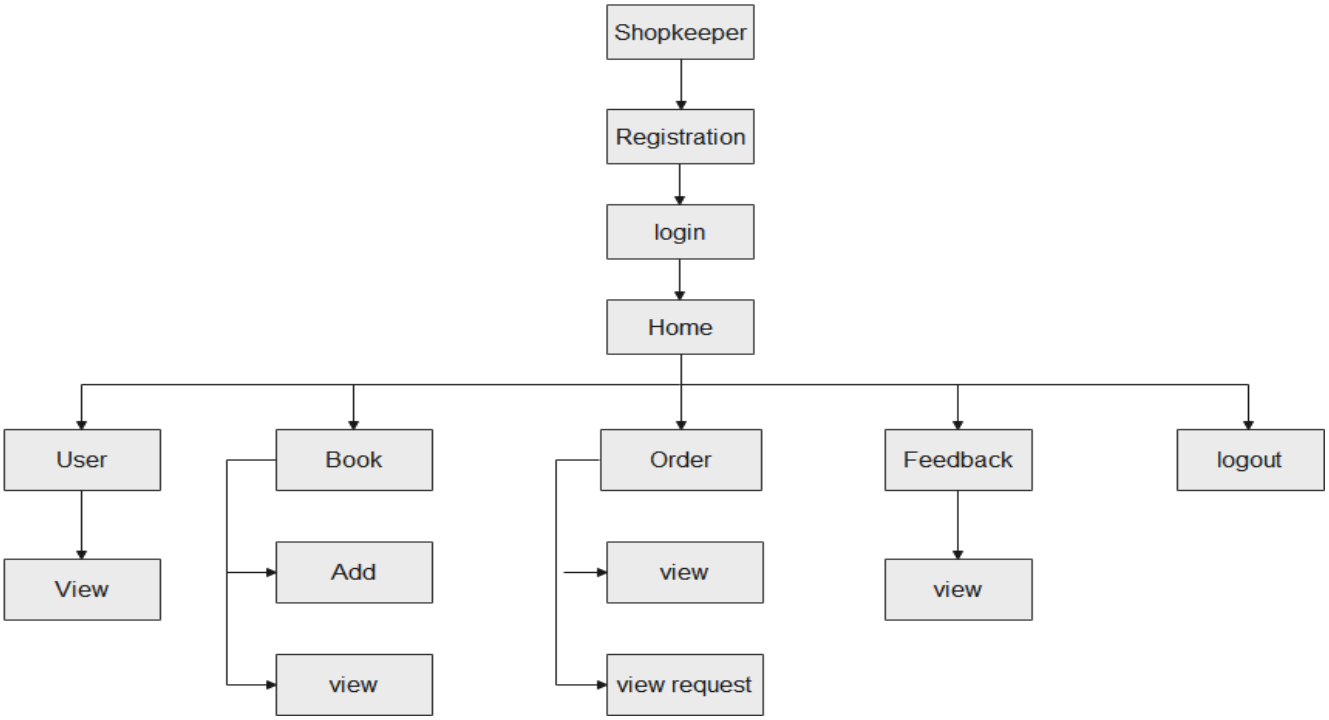


Figure 5.2.2 Structure chart for Shopkeeper

c) User Module

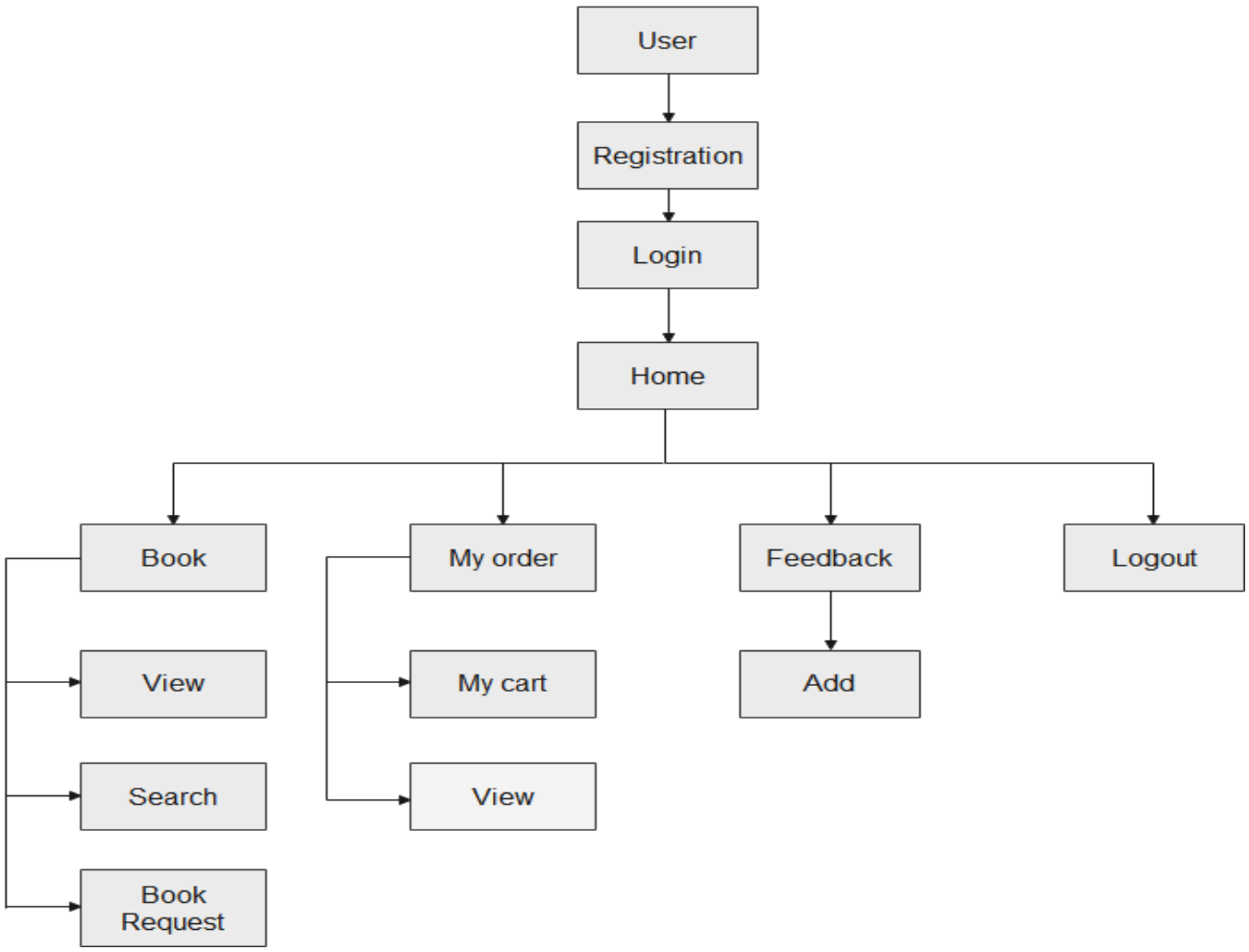


Figure 5.2.3 Structure chart for User

5.3. Modular Decomposition of the System

5.3.1 Admin Module

5.3.1.1 Login page:

a) Input : Email id, Password.

b) Procedure Details :

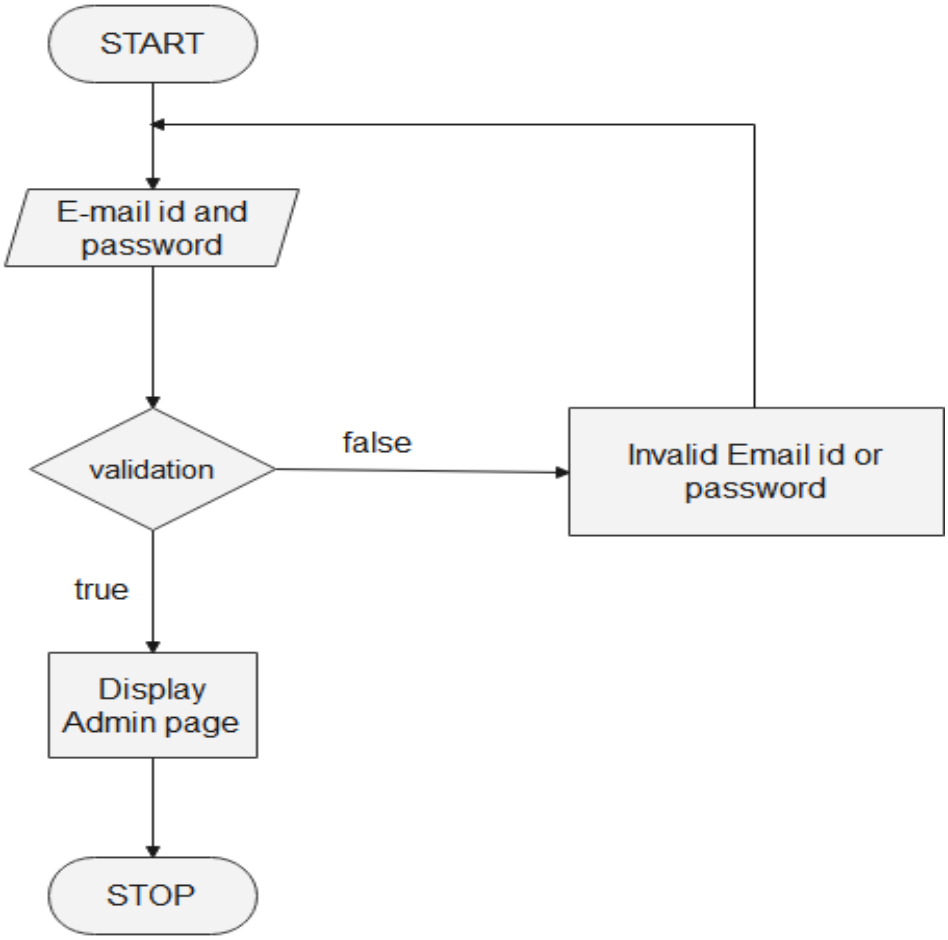


Figure 5.3.1.1 Admin login

c) File I/O Interface : Graphical user interface to login as Admin.

d) Output : Entered Email id and password will be checked for validity if it is valid, Admin will be redirected to Admin page.

5.3.1.2 View Shopkeeper

a) Input : Click on shopkeeper management.

b) Procedure Details :

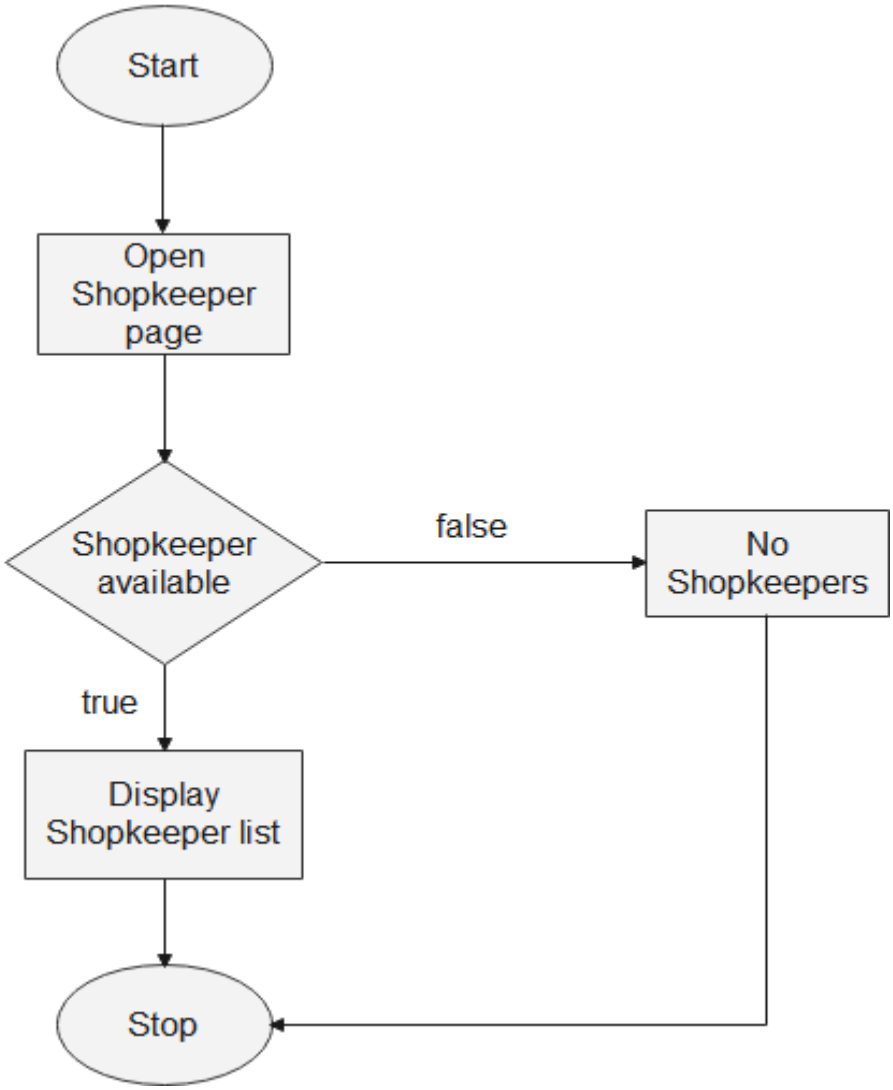


Figure 5.3.1.2 View shopkeeper

c) File I/O Interface : Graphical user interface to enable and disable shopkeepers data are stored in the shopkeeper table.

d) Output : List of shopkeepers along with their details are displayed here.

5.3.1.3 View User

a) Input : Click on user management.

b) Procedure Details :

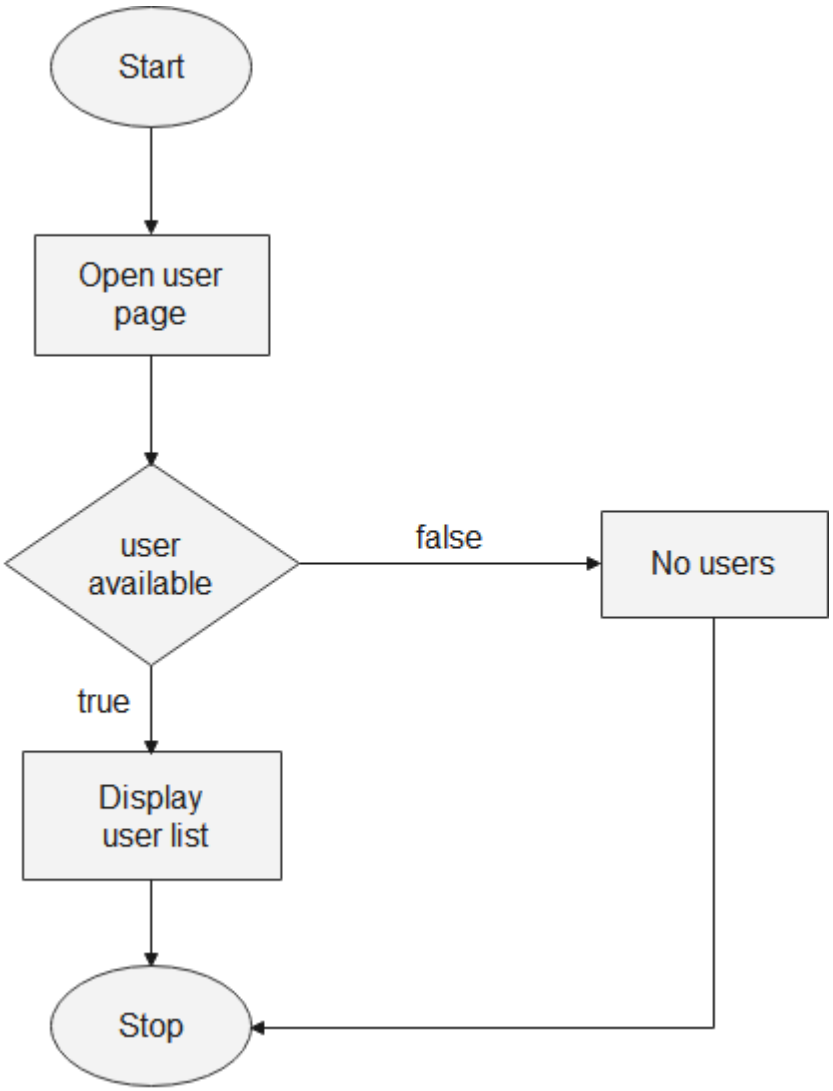


Figure 5.3.1.3 View User

c) File I/O Interface : Graphical user interface to enable and disable Users.Details are stored in the User table.

d) Output : List of users along with their details are displayed here.

5.3.2 Shopkeeper Module

5.3.2.1 Shopkeeper Registration

a) Input : Shop Name ,Contact No ,Email Id, password, address.

b) Procedure details :

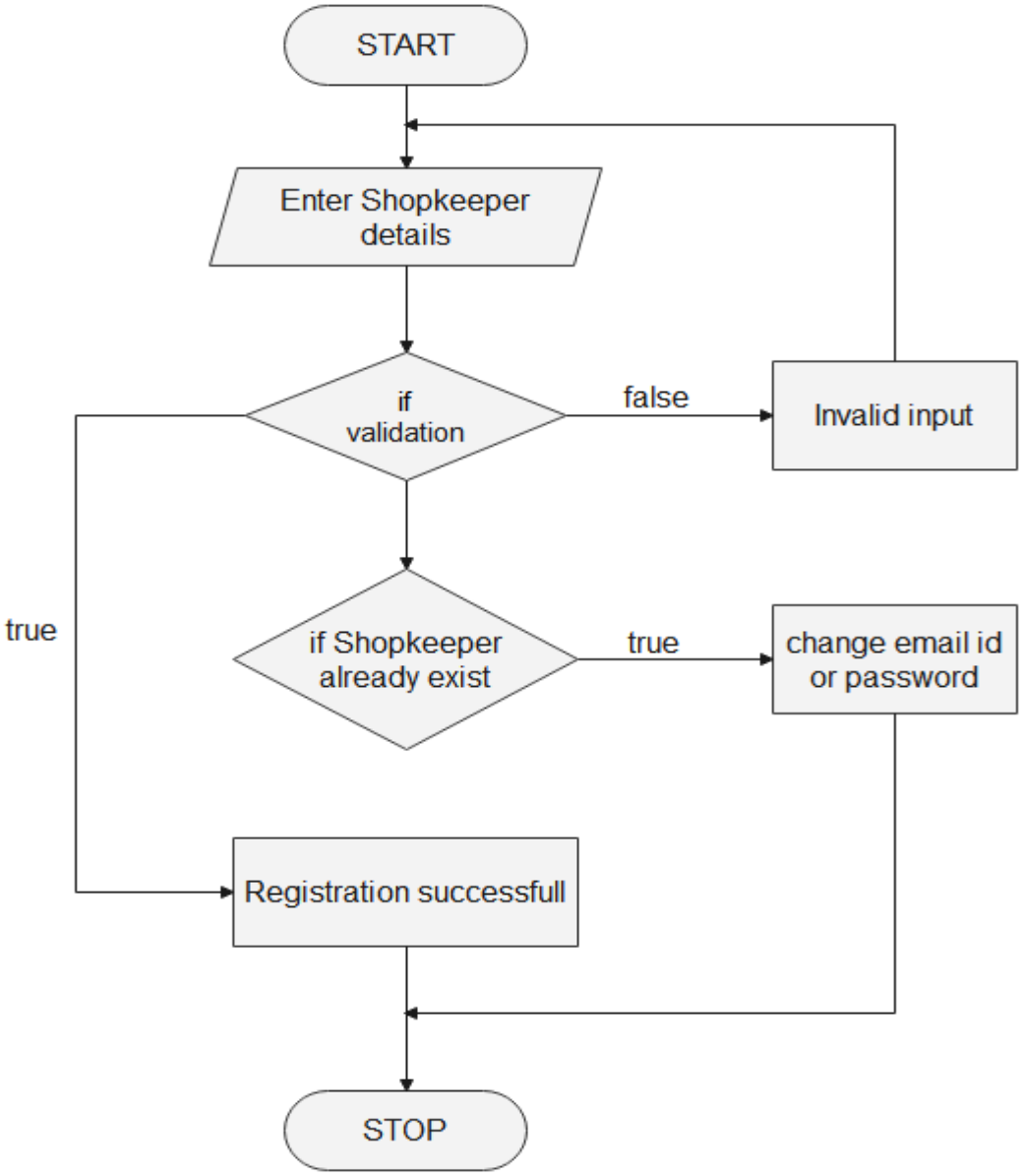


Figure 5.3.2.1 Shopkeeper Registration

c) File I/O Interface : Graphical user interface to enter Shopkeeper Details. Data Stored in Shopkeeper table. .

d) Output : Entered details will be checked for validity if it is valid, then Successfully registered.

5.3.2.2 Shopkeeper Login

a) Input : Email id, Password.

b) Procedure Details :

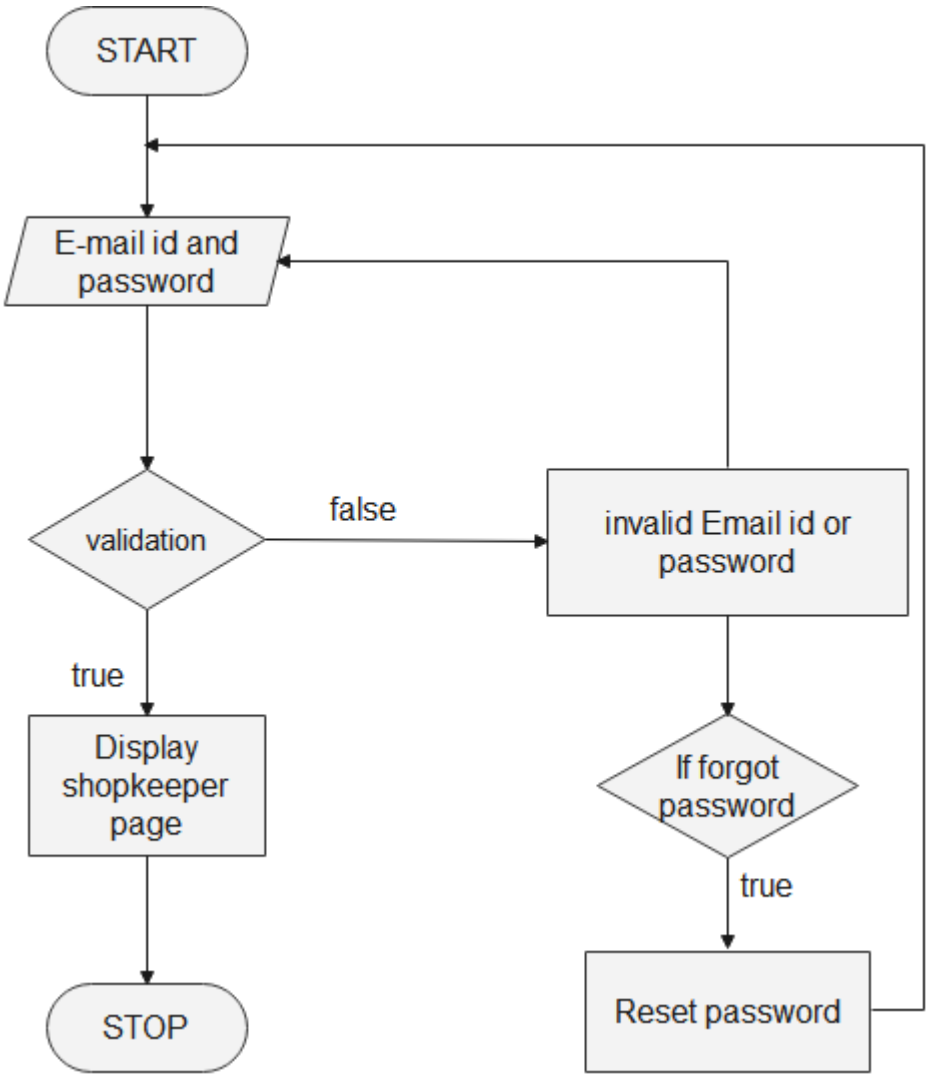


Figure 5.3.2.2 Shopkeeper Login

c) File I/O Interface : Graphical user interface to login as Shopkeeper.

d) Output : Entered Email id and password will be checked for validity if it is valid, Shopkeeper will be redirected to Shopkeeper page.

5.3.2.3 View Users

c) Input : Click on users

d) Procedure Details :

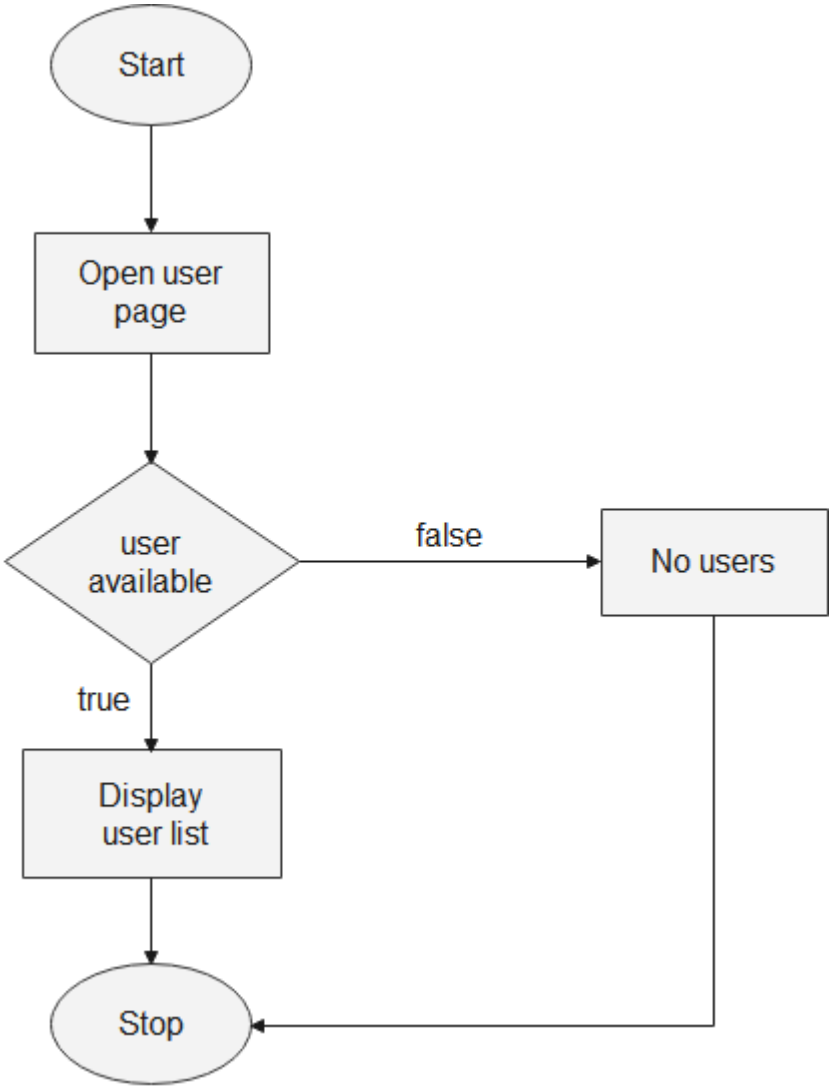


Figure 5.3.2.3 View Users

a) File I/O Interface : Graphical user interface to view the list of registered users.The data is stored in User table.

b) Output : The registered users are displayed here.

5.3.2.4 Add Book

a) Input : Book name,Author Name, Publisher ,Choose category,description, Book price,no of copies,choose book image.

b) Procedure Details :

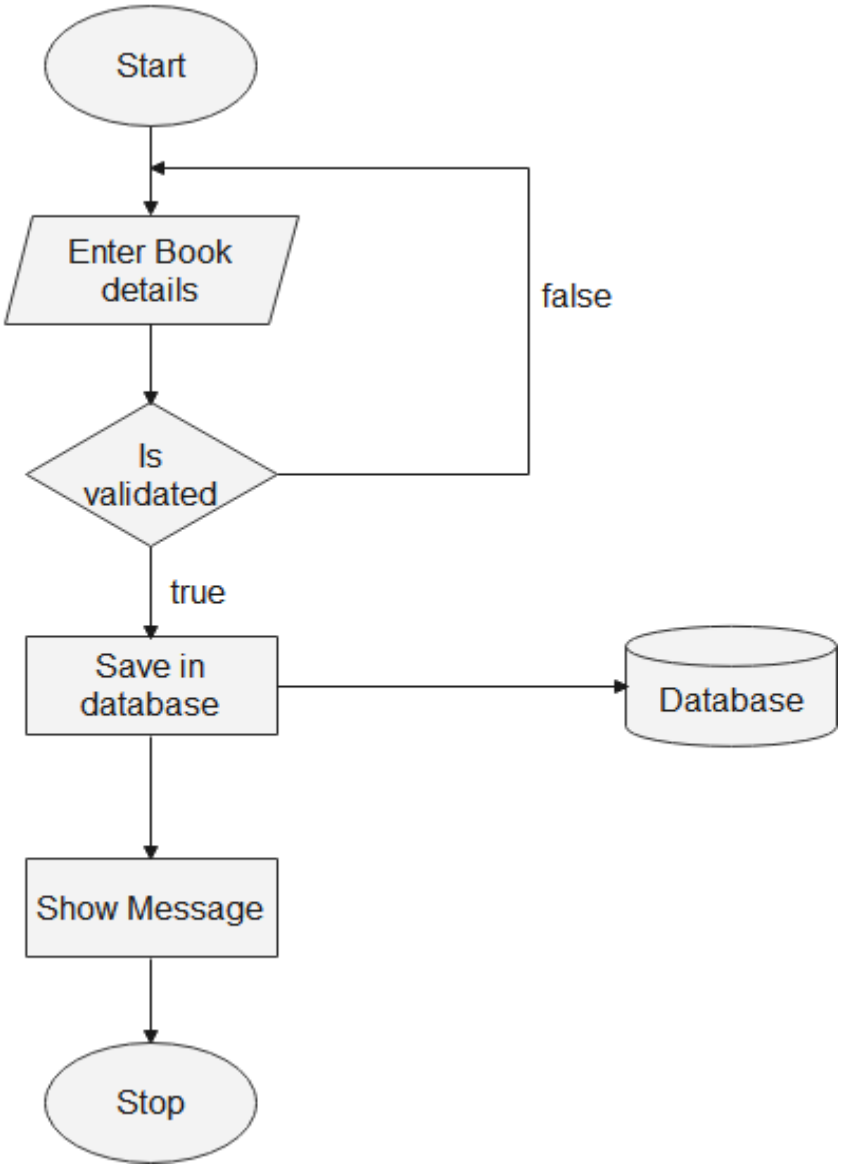


Figure 5.3.2.4 Add book

c) File I/O Interface : Graphical user interface to enter Book details. Data is Stored in book Table.

d) Output : Entered Book details will be checked for validity, if it is valid system shows completion message.

5.3.2.5 View Book orders

c) Input : Click on view orders.

d) Precedure details :

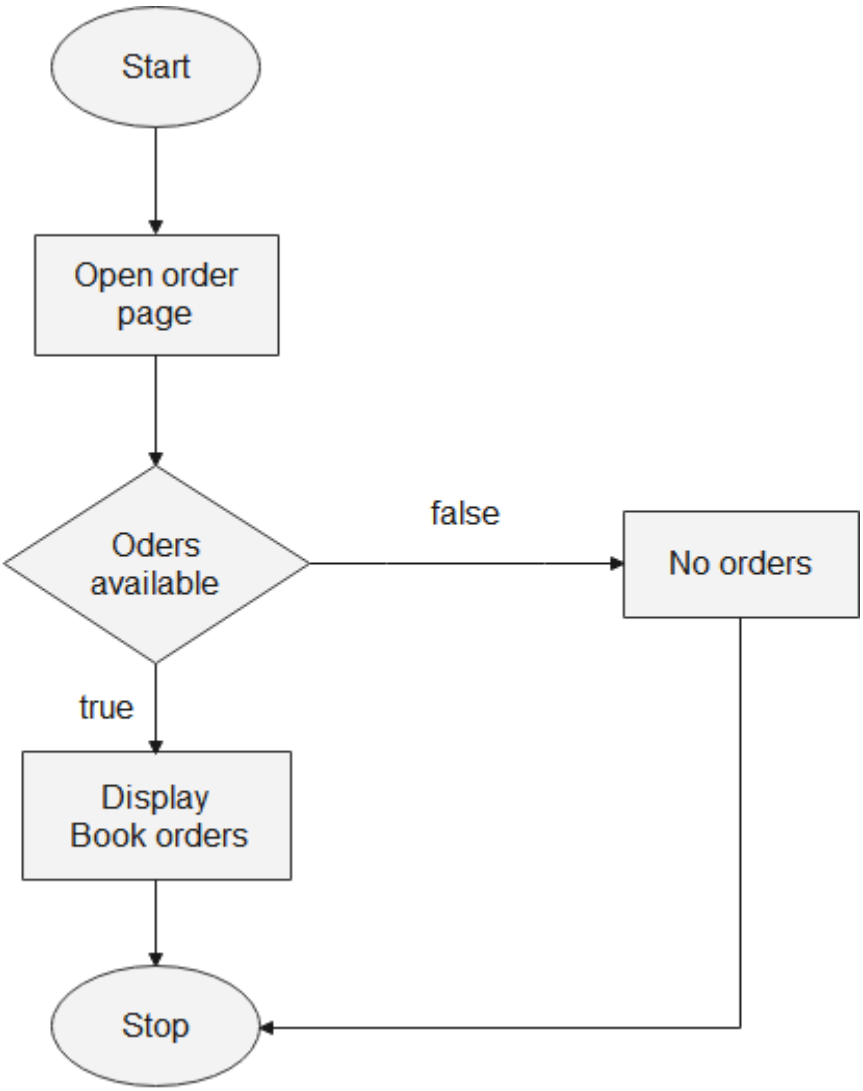


Figure 5.3.2.5 View orders

e) File I/O Interface : Graphical user interface to view book orders made by the users.The data is stored in order table.

f) Output : Orders made by the user is displayed here.

5.3.2.6 View Feedback

a) Input : Click on feedback.

b) Procedure details :

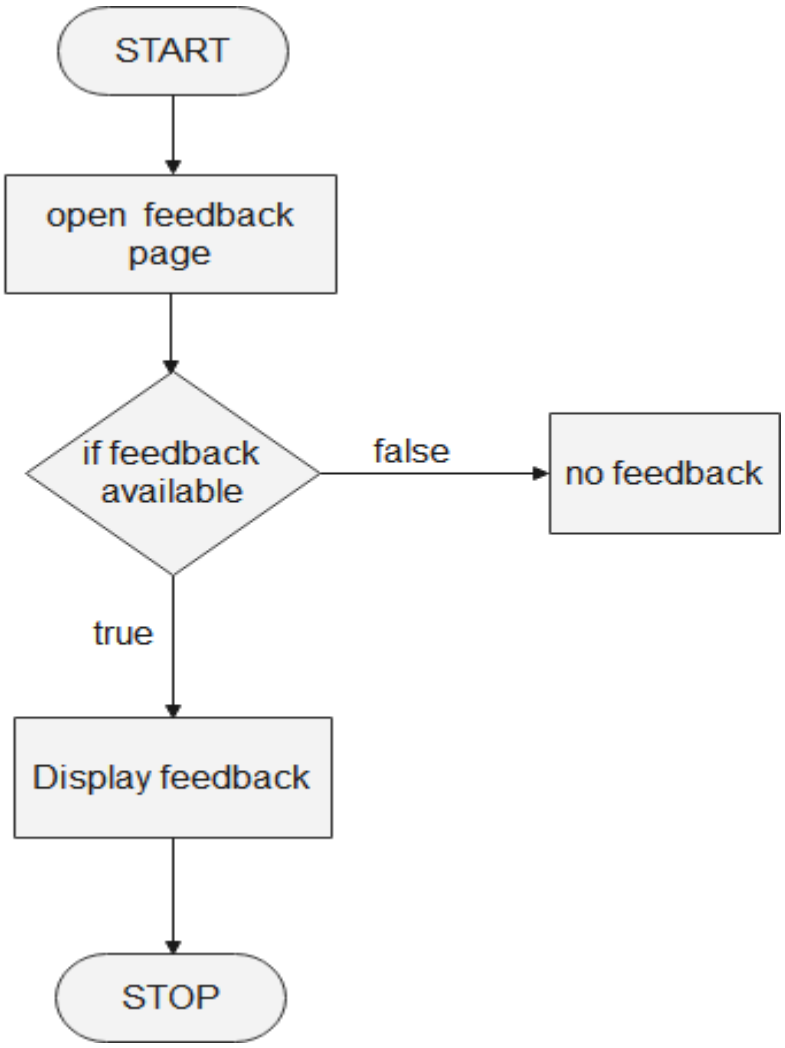


Figure 5.3.2.6 View Feedback

c) File I/O Interface : Graphical user interface to view feedback given by the user regarding book. The data is stored in feedback table.

d) Output : Feedbacks which are given by users will be displayed here.

5.3.3 User Module

5.3.3.1 User Registration

a) Input : Name. contact no, Email-id, password, address.

b) Preocedure details :

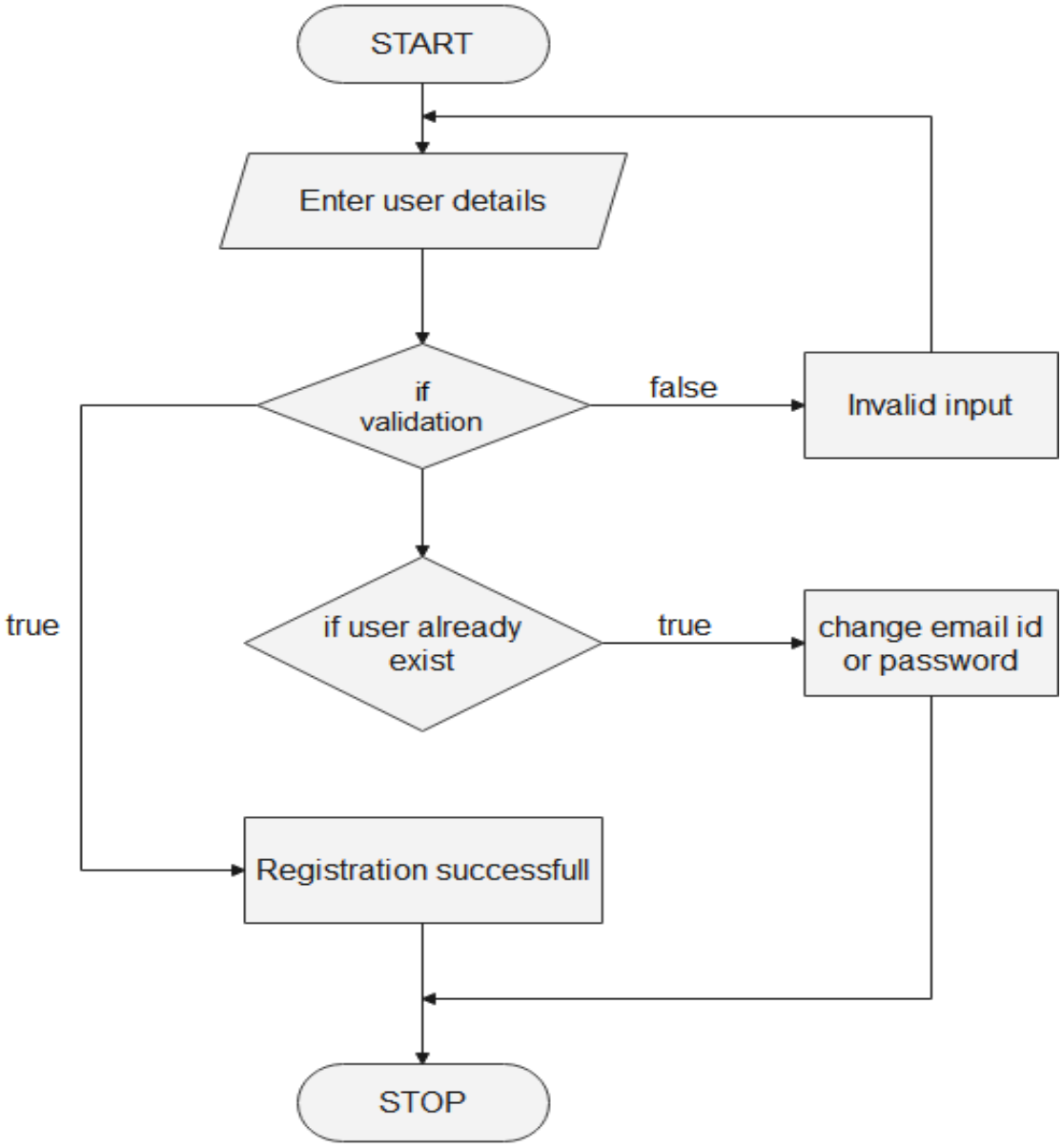


Figure 5.3.3.1 user Registration

a) File I/O I nterface : Graphical user interface to enter user Details. Data is stored in the user table.

b) Output : Entered details will be checked for validity if it is valid, then successfully registered.

5.3.3.2 User login

a) Input : Email_id, password.

b) Procedure details :

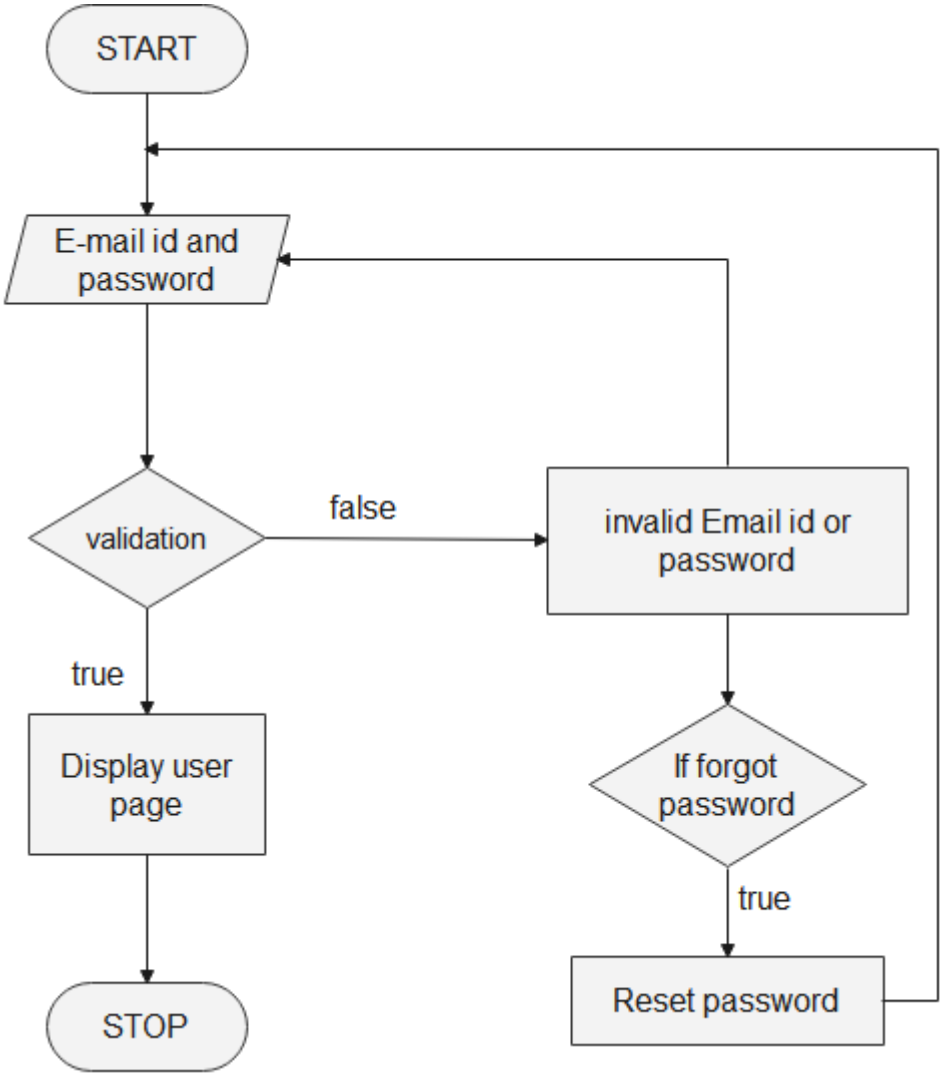


Figure 5.3.3.2 user Login

c) File I/O Interafce : Graphical user interface to login as user .

d) Output : Entered Email id and password will be checked for validity if it is valid, User will be redirected to Home page.

5.3.3.3 View Book

a) Input : Enter Book name

b) Procedure Details :

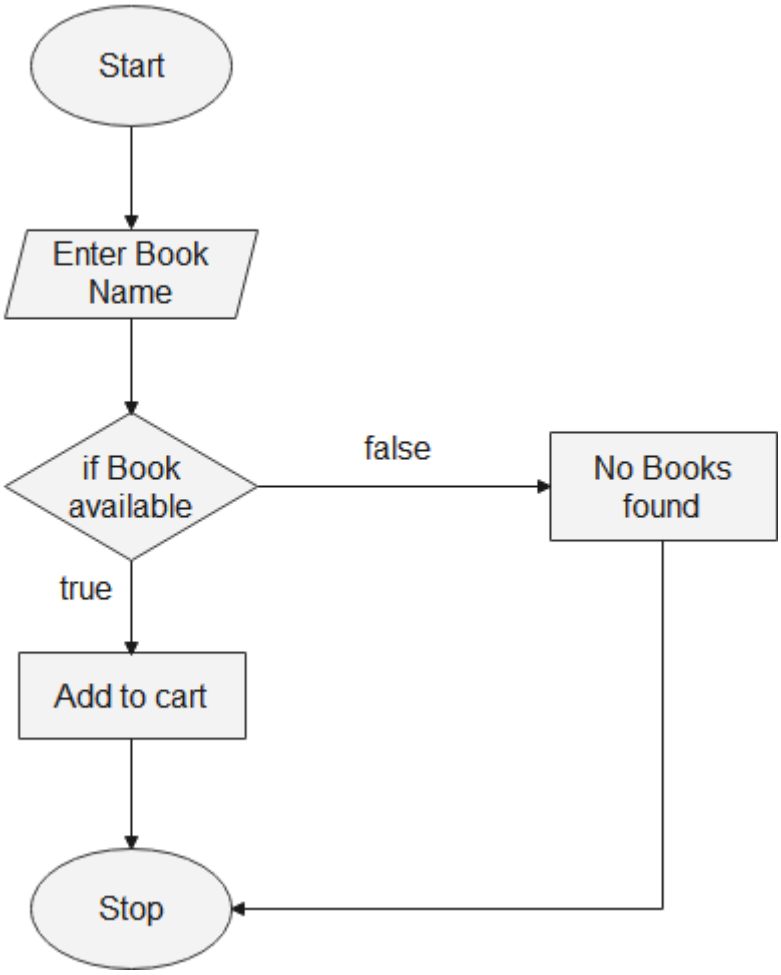


Figure 5.3.3.3 View Book

a) File I/O Interface : Graphical User interface to search and view books.

b) Output : The books which are available in the store are displayed here.

5.3.3.4 User cart

a) Input : Click on add to cart.

b) Procedure details :

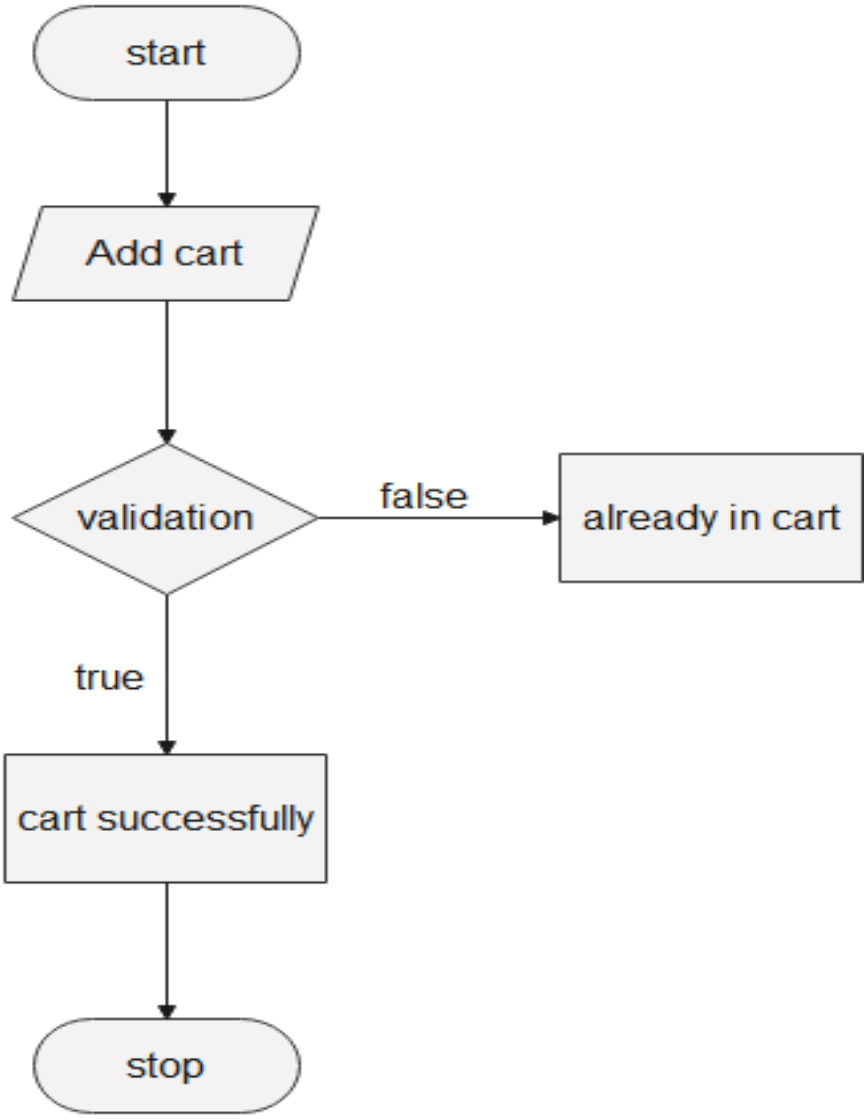


Figure 5.3.3.4 user cart

c) File I/O Interface : Graphical User interface to add book to the cart.

d) Output : The books which are added to the cart are displayed here.

5.3.3.5 Order Book

c) Input : Click on My Cart.

d) Procedure Details :

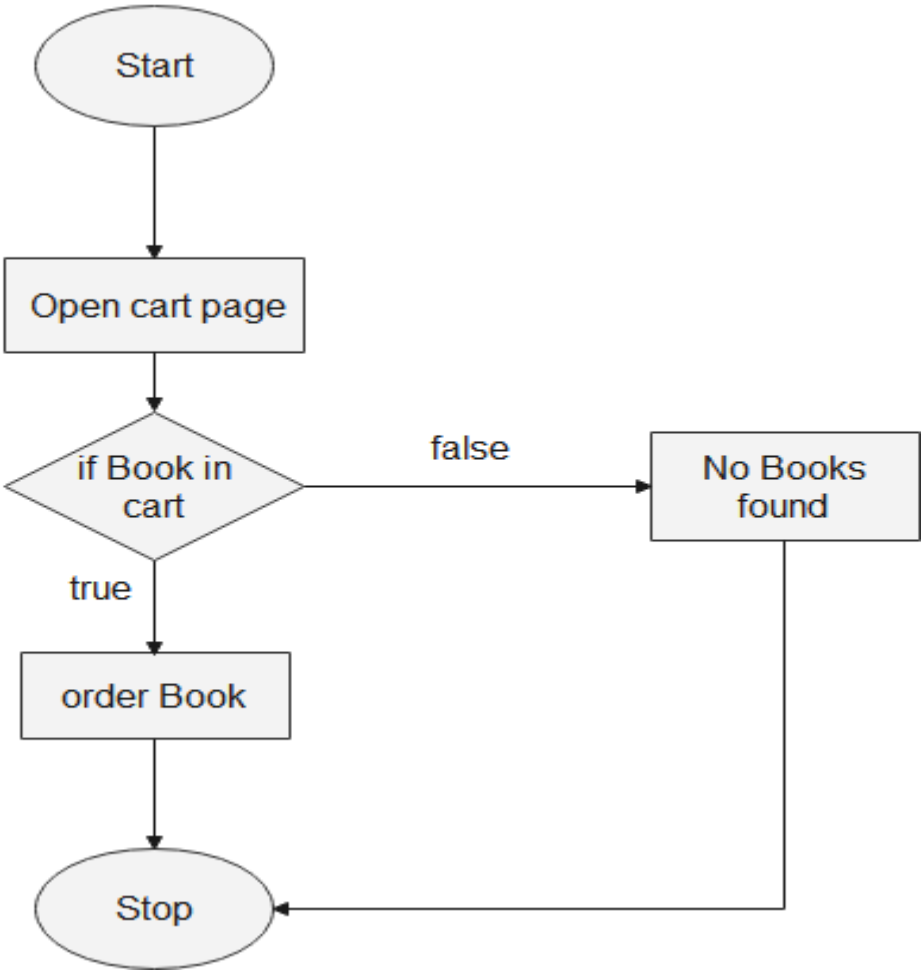


Figure 5.3.3.5 Order Book

e) File I/O Interface : Graphical user interface to order books based on different category from the cart.

f) Output : The order is added and stored in database.The data is stored in order Table.

5.3.3.6 Write FeedBack

a) Input : Book name, description.

b) Procedure Details :

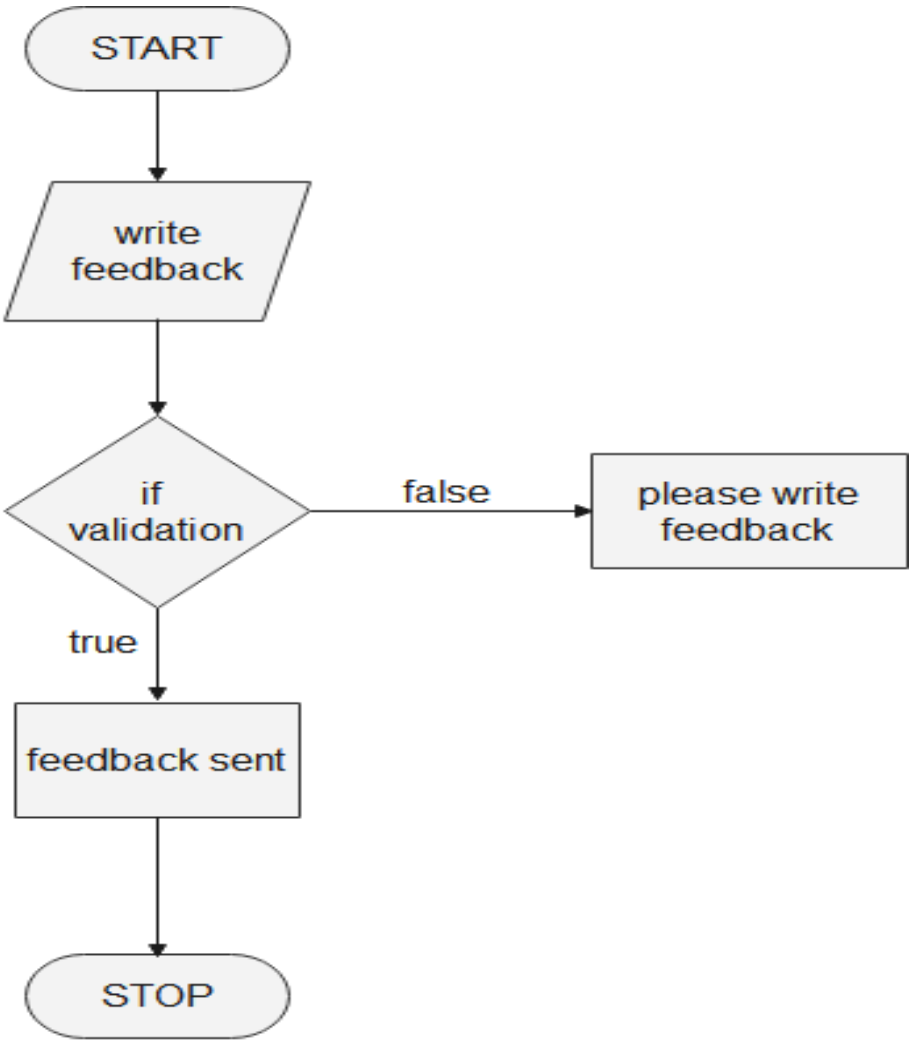


Figure 5.3.3.6 write Feedback

c) File I/O Interface : Graghical user interface for user to write the feedback of the book purchased.The data is stored in Feedback table.

d) Output : The given feedback is stored and displayed to the shopkeeper.

CHAPTER - 6

CODING

6.1 Introduction

The goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in the given programming language, which can be executed by a computer and that performance the computation performed by the design.

6.2 Database connection

The function to connect to MySQL is called MySQL connect. This function returns a resource which is a pointer to the database connection.

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'online_book_store');
?>
```

6.3 Authorization / Authentication

Authentication is the process of verifying who you are. When you log on to a PC with a user name and password you are authenticating.

Authorization is the process of verifying that you access to something gaining access to a resource.

Login & register form for Admin, Shopkeeper and user

- **Admin Login**

```
<?php
session_start();
$error = '';
if (isset($_GET['error'])) {
   $error = $_GET['error'];
}
if (isset($_POST['command'])) {
$userName = $_POST['username'];
$password = $_POST['password'];
 if ('admin' == $userName && 'admin' == $password)
 $_SESSION['admin'] = true;
 $_SESSION['login'] = true;
 header('location: home.php');
```

```
} else {
$error = "Invalid login credentials!";
}
}
?>
```

- **Shopkeeper Registration :**

```
<?php
require_once './../db/db.php';
$con = new DB();
$error = '';
if (isset($_GET['error'])) {
$error = $_GET['error'];
}
if (isset($_POST['command'])) {
$name = $_POST['name'];
$contact_no = $_POST['contact'];
$email = $_POST['email'];
$address = $_POST['address'];
$password = $_POST['password'];
$query = "SELECT * FROM `shop_keeper` WHERE email_id='$email'";
$result = $con->executeSelect($query);
if (empty($result)) {
$sql = "INSERT INTO `shop_keeper`(shop_name,contact_no,email_id,address,password)
VALUES('$name','$contact_no','$email','$address','$password')";
 $res = $con->executeInsertAndGetId($sql);
 if ($res > 0) {  ?>
<script> alert("Registered successfully");
 window.location = 'login.php';
 </script>
 <?php } else { ?>
 <script>   alert("Something went wrong!");
 window.location = 'register.php';</script>
 <?php
  }
  } else {
  ?>
  <script>   alert("You already registered with this Email Id!");
  window.location = 'register.php';</script>
  <?php
 }
}
?>
```

- **Shopkeeper Login**

```
<?php
session_start();
require_once './../db/db.php';
$con = new DB();
$error = '';
if (isset($_GET['error'])) {
$error = $_GET['error'];
}
if (isset($_POST['command'])) {
$email_id = $_POST['email_id'];
$password = $_POST['password'];
$sql = "SELECT * FROM `shop_keeper` WHERE email_id='$email_id' AND
password='$password'";
$res = $con->executeSelect($sql);
if (!empty($res)) {
 $sql = "SELECT * FROM `shop_keeper` WHERE email_id='$email_id' AND
password='$password' AND is_enabled=1";
$result = $con->executeSelect($sql);
 if (!empty($result)) {
 $_SESSION['login'] = true;
 $_SESSION['shopkeeper_id'] = $res[0]['shop_keeper_rid'];
 $_SESSION['shopkeeper_name'] = $res[0]['shop_name'];
  header('location: home.php');
  } else {
 $error = "Invalid user!";
  }
  } else {
  $error = "Invalid login credentials!";
  }
}
?>
```

- **User Registration** :

```
<?php
require_once './../db/db.php';
$con = new DB();
$error = '';
if (isset($_GET['error'])) {
$error = $_GET['error'];
}
if (isset($_POST['command'])) {
$name = $_POST['name'];
$contact_no = $_POST['contact'];
```

```
$email = $_POST['email'];
$address = $_POST['address'];
$password = $_POST['password'];
$query = "SELECT * FROM user WHERE email_id='$email'";
$result = $con->executeSelect($query);
if (empty($result)) {
$sql = "INSERT INTO user(name,contact_no,email_id,address,password)
VALUES('$name','$contact_no','$email','$address','$password')";
 $res = $con->executeInsertAndGetId($sql);
  if ($res > 0) {
  ?>
  <script> alert("Registered successfully");
  window.location = 'login.php';
 </script>
  <?php } else { ?>
  <script>   alert("Something went wrong!");
   window.location = 'register.php';</script>
    <?php
   }
    } else {
    ?>
   <script>   alert("You already registered with this Email Id!");
   window.location = 'register.php';</script>
   <?php
    }
}
?>
```

- **User Login**

```
?php
session_start();
require_once './../db/db.php';
$con = new DB();
$error = '';
if (isset($_GET['error'])) {
$error = $_GET['error'];
}
if (isset($_POST['command'])) {
$email_id = $_POST['email_id'];
$password = $_POST['password'];
$sql = "SELECT * FROM user WHERE email_id='$email_id' AND password='$password'";
$res = $con->executeSelect($sql);
if (!empty($res)) {
```

```
$sql = "SELECT * FROM user WHERE email_id='$email_id' AND password='$password' AND
is_enabled=1";
$result = $con->executeSelect($sql);
if (!empty($result)) {
$_SESSION['login'] = true;
$_SESSION['user_id'] = $res[0]['user_id'];
 $_SESSION['user_name'] = $res[0]['name'];
 header('location: view_book.php');
} else {
$error = "Invalid user!";
}
} else {
$error = "Invalid login credentials!";
}
}
?>
```

6.4 Data store / retrieval/ update

- **Code for Admin module**

View shopkeeper

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$query = "SELECT * FROM `shop_keeper`";
$con = new DB();
$shop_keeper = $con->executeSelect($query);
?>
<html>
<?php
 ?>
<body>
 <br>
 <div class="container">
  <div class="row justify-content-center">
  <div class="col-md-12">
<div class="card">
<div class="card-header">
<h4 class="text-center"> Shopkeeper List</h4>
  </div>
 <div class="card-body">
<table class="table table-bordered table-sm">
<thead>
<tr class="text-center">
```

```
<th>#</th>
<th>Shop Name</th>
 <th>Contact Number</th>
 <th>Email Id</th>
  <th>Password</th>
  <th>Address</th>
   <th>Action</th>
  </tr>
  </thead>
  <tbody>
<?php
 $i = 0;
 foreach ($shop_keeper as $row) {
?>
<tr style="text-align: center;">
<td><?php echo ++$i; ?></td>
 <td><?php echo "$row[shop_name]" ?></td>
 <td><?php echo"$row[contact_no]" ?></td>
 <td><?php echo"$row[email_id]" ?></td>
 <td><?php echo "$row[password]" ?></td>
   <td><?php echo"$row[address]" ?></td>
   <?php
 $state = $row['is_enabled'];
 if ($state == 0) {
 ?>
 <td><button type="button" class="btn btn-secondary btn-sm ml-3 px-2 pr-3" id="btn_enable"
onclick="enable_shopkeeper(<?php echo "$row[shop_keeper_rid]" ?>);">Enable</button></td>
<?php } else {
  ?>
 <td><button type="button" class="btn btn-primary btn-sm ml-3 px-2" id="btn_disable"
onclick="disable_shopkeeper(<?php echo "$row[shop_keeper_rid]" ?>);">Disable</button></td>
   <?php }
?>
 </tr>
 <?php
  }
 if ($i == 0) {
  ?>
 <tr>
 <td colspan="100%" class=" text-center">
No records
</td>
</tr>
```

```
<?php }
?>
</tbody>
</table>
</div>
</div>
</div>
 </div>
 </div>
 <?php require_once '../include/footer.php'; ?>
  <script src="../static/js/shopkeeper_action.js" type="text/javascript"></script>
  </body></html>
```

View User

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$query = "SELECT * FROM `user`";
$con = new DB();
$user = $con->executeSelect($query);
?>
<html>
<?php
 ?>
 <body>
 <br>
 <div class="container">
 <div class="row justify-content-center">
 <div class="col-md-12">
  <div class="card">
   <div class="card-header">
   <h4 class="text-center">User List</h4>
    </div>
   <div class="card-body">
 <table class="table table-bordered table-sm">
<thead>
<tr class="text-center">
 <th>#</th>
  <th>Name</th>
<th>Contact Number</th>
<th>Email Id</th>
<th>Password</th>
<th>Address</th>
```

```
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
 $i = 0;
foreach ($user as $row) {
?>
<tr style="text-align: center;">
<td><?php echo ++$i; ?></td>
<td><?php echo "$row[name]" ?></td>
<td><?php echo"$row[contact_no]" ?></td>
<td><?php echo"$row[email_id]" ?></td>
<td><?php echo "$row[password]" ?></td>
 <td><?php echo"$row[address]" ?></td>
 <?php
 $state = $row['is_enabled'];
 if ($state == 0) {
?>
<td><button type="button" class="btn btn-secondary btn-sm ml-3 px-2 pr-3" id="btn_enable"
onclick="enable_user(<?php echo "$row[user_id]" ?>);">Enable</button></td>
<?php } else {
 ?>
<td><button type="button" class="btn btn-primary btn-sm ml-3 px-2" id="btn_disable"
onclick="disable_user(<?php echo "$row[user_id]" ?>);">Disable</button></td>
<?php }
 ?>
</tr>
 <?php
}
if ($i == 0) {
?>
<tr>
<td colspan="100%" class="text-center">
No records
</td>
</tr>
<?php }
?>
</tbody>
</table>
</div>
</div>
```

```
</div>
</div></div>
<?php require_once '../include/footer.php'; ?>
<script src="../static/js/user_action.js" type="text/javascript"></script>
 </body></html>
```

6.5 Data validation

```
<?php
session_start();
require_once '../shop_keeper/Response.php';
require_once './../db/db.php';
require_once '../include/util.php';
$con = new DB();
$response = new Response();
try {
if (isset($_POST['command']))
$command = $_POST['command'];
if ('enableUser' == $command) {
 $id = $_POST['id'];
 $sql = "UPDATE `user` SET is_enabled=1 WHERE user_id=$id";
$res = $con->executeUpdate($sql);
if ($res > 0) {
$response->success("Completed successfully...");
 } else {
 throw new Exception("Something went wrong...");
}
} else if ('disableUser' == $command) {
$id = $_POST['id'];
$sql = "UPDATE `user` SET is_enabled=0 WHERE user_id=$id";
$res = $con->executeUpdate($sql);
 if ($res > 0) {
$response->success("Completed successfully...");
} else {
 throw new Exception("Something went wrong...");
 }
 } else if ('enableShopkeeper' == $command) {
 $id = $_POST['s_id'];
 $sql = "UPDATE `shop_keeper` SET is_enabled=1 WHERE shop_keeper_rid=$id";
$res = $con->executeUpdate($sql);
if ($res > 0) {
$response->success("Completed successfully...");
} else {
 throw new Exception("Something went wrong...");
  }
```

```
} else if ('disableShopkeeper' == $command) {
$id = $_POST['s_id'];
 $sql = "UPDATE `shop_keeper` SET is_enabled=0 WHERE shop_keeper_rid=$id";
$res = $con->executeUpdate($sql);
 if ($res > 0) {
$response->success("Completed successfully...");
} else {
 throw new Exception("Something went wrong...");
 }
}
} else if (isset($_GET['command'])) {
$command = $_GET['command'];
}
} catch (Exception $ex) {
 $response->error($ex->getMessage());
}
$response->writeResponse();
```

Logout :

```
<?php
session_start();
session_destroy();
header("location: ../index.php");
```

- **Code for shopkeeper module**

  Add Book

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$category = "SELECT * FROM `category`";
$con = new DB();
$book_category = $con->executeSelect($category);
?>
<html>
<?php
?>
<body >
<div class="container-fluid" style="background-image:
url('../static/images/online2.jpg');background-repeat: no-repeat;background-size: cover;background-
position: bottom;">
<div class="row justify-content-center">
 <div class="col-md-5">
<div class="card my-2">
<div class="card-body">
```

```
<h3 class="text-center">Add Book</h3>
<form action="action.php" method="post" id="formAddBook" enctype="multipart/form-data">
<input type="hidden" name="command" value="addBook"/>
<div class="form-group">
<input class="form-control" id="book_name" name="book_name" placeholder="Book Name"
autocomplete="off"/>
</div>
<div class="form-group">
<input class="form-control" id="author_name" name="author_name" placeholder="Author Name"
autocomplete="off"/>
</div>
<div class="form-group">
<input class="form-control" id="publisher_name" name="publisher_name"
placeholder="Publisher" autocomplete="off"/>
</div>
<div class="form-group">
<select class="form-control" name="book_category" id="book_category">
<option>--Choose Category--</option>
 <?php foreach ($book_category as $row) { ?>
<option value="<?php echo $row['category_id']; ?>"><?php echo $row['category_name'];
?></option>
<?php }
?>
</select>
</div>
<div class="form-group">
<textarea class="form-control" id="description" name="description rows="3"
placeholder="Description" autocomplete="off"></textarea>
</div><div class="form-group">
<input class="form-control" type="number" id="book_price" name="book_price"
placeholder="Book Price" autocomplete="off"
</div>
<div class="form-group">
<input class="form-control" type="number" id="book_copies" name="book_copies"
placeholder="No. Of Book Copies" autocomplete="off"/>
</div>
<div class="form-group">
<input type="file" class="form-control" id="book_image" name="book_image"/>
</div>
<div class="form-group">
<button class="btn btn-dark btn-block" id="btnAddBook">Add Book</button>
</div>
</form>
```

```
</div>
</div>
</div>
</div>
</div>
<?php require_once '../include/footer.php'; ?>
<script src="../static/js/add_book.js" type="text/javascript"></script>
</body></html>
```

Update books

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$id = $_GET['id'];
$query = "SELECT * FROM book AS b JOIN `category` AS c ON c.category_id=b.book_category
WHERE b.book_rid='$id'";
$con = new DB();
$get_books = $con->executeSelect($query);
$bk_category_id = $get_books[0]['book_category'];
$query2 = "SELECT * FROM `category` WHERE NOT category_id='$bk_category_id'";
$bk_category = $con->executeSelect($query2);
?>
<html>
<?php
?>
<body>
<div class="container mt-2">
<div class="row justify-content-center">
<div class="col-md-5">
<div class="card">
 <div class="card-body">
<h3 class="text-center">Book Details</h3>
<hr>
<form action="action.php" method="post" id="formUpdateBook" enctype="multipart/form-data">
<?php foreach ($get_books as $row) { ?>
<input type="hidden" name="command" value="updateBook"/>
<input type="hidden" name="update_book_id" id="update_book_id" value="<?php echo
$row['book_rid']; ?>">
<div class="form-group">
<input class="form-control" id="up_book_name" name="up_book_name"
placeholder="Book Name" autocomplete="off" value="<?php echo $row['book_name']; ?>" />
</div>
<div class="form-group">
```

```
<input class="form-control" id="up_author_name" name="up_author_name" placeholder="Author Name" autocomplete="off" value="<?php echo $row['author_name']; ?>"  />
</div>
<div class="form-group">
<input class="form-control" id="up_publisher_name" name="up_publisher_name" placeholder="Publisher" autocomplete="off" value="<?php echo $row['publisher_name']; ?>"
</div>
<div class="form-group">
<select class="form-control" name="up_book_category" id="up_book_category">
<option value="<?php echo $row['category_id']; ?>"selected><?php echo $row['category_name']; ?></option>
<option>--Choose Category--</option>
<?php foreach ($bk_category as $bk) { ?>
<option value="<?php echo $bk['category_id']; ?>"><?php echo $bk['category_name']; ?></option>
<?php } 
?>
</select>
</div>
<div class="form-group">
<textarea class="form-control" id="up_description" name="up_description" rows="3" placeholder="Description" autocomplete="off"><?php echo $row['description']; ?></textarea>
</div>
<div class="form-group">
<input class="form-control" type="number" id="up_book_price" name="up_book_price" placeholder="Book Price" autocomplete="off" value="<?php echo $row['book_price']; ?>" />
</div>
<div class="form-group">
<input class="form-control" id="up_book_copy" name="up_book_copy" type="number" placeholder="No .of Book Copies"autocomplete="off"value="<?php echo $row['book_copy']; ?>"
/>
</div>
<div class="form-group">
<button class="btn btn-dark btn-block" id="btnUpdateBook">Update</button>
</div>
<?php } 
?>
</form>
</div>
</div>
</div>
</div>
</div>
```

```
<?php require_once '../include/footer.php'; ?>
<script src="../static/js/add_book.js" type="text/javascript"></script>
</body></html>
```

View orders

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$con = new DB();
$employee = $con->executeSelect($query);
?>
<html>
<?php
?>
<body>
<br><div class="container">
<div class="row justify-content-center">
<div class="col-md-12">
<div class="card">
<div class="card-header">
<h4 class="text-center">Order List</h4>
</div>
<div class="card-body">
<table class="table table-bordered table-sm">
<thead>
<tr class="text-center">
<th>#</th>
<th>Book Buyer</th>
<th>Contact Number</th>
<th>Book Name</th>
<th>Book Copies</th>
<th>Price(in Rs.)</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
<?php
$i = 0;
foreach ($employee as $row) {
?>
<tr style="text-align: center;">
<td><?php echo ++$i; ?></td>
<td><?php echo "$row[name]" ?></td>
```

```
<td><?php echo"$row[contact_no]" ?></td>
<td><?php echo"$row[book_name]" ?></td>
<td><?php echo"$row[copies]" ?></td>
<td>Rs.<?php echo "$row[total_cost]" ?>/-</td>
<?php
$state = $row['is_paid'];
if ($state == 'no') {
?>
<td><button type="button" class="btn-danger btn-sm" disabled>Not Paid</button></td>
<?php } else {
?>
<td><button type="button" class="btn-success btn-sm px-3 disabled">Paid</button></td>
<?php }
?>
</tr>
<?php
}
if ($i == 0) {
?>
<tr>
<td colspan="100%" class=" text-center">No records</td>
</tr>
<?php }
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<?php require_once '../include/footer.php'; ?>
</body></html>
```

View feedback

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$shopkeep_id = $_SESSION['shopkeeper_id'];
$query = "SELECT * FROM feedback AS f JOIN `user` AS u ON u.user_id=f.user_id "."JOIN
book AS b ON b.book_rid=f.book_id WHERE b.shopkeeper_id='$shopkeep_id' ORDER BY f.date
DESC";
$con = new DB();
```

```
$feedback = $con->executeSelect($query);
?>
<html>
<?php
?>
<body>
<br><div class="container">
<div class="row justify-content-center">
<div class="col-md-12">
<div class="card">
<div class="card-header">
<h4 class="text-center"> Feedback List</h4>
</div>
<div class="card-body">
<table class="table table-bordered table-sm">
<thead>
<tr class="text-center">
<th>#</th>
<th>Book Buyer</th>
<th>Contact No.</th>
<th>Address</th>
 <th>Book Name</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<?php
$i = 0;
foreach ($feedback as $row) {
?>
<tr style="text-align: center;">
<td><?php echo ++$i; ?></td>
<td><?php echo "$row[name]" ?></td>
<td><?php echo "$row[contact_no]" ?></td>
<td><?php echo "$row[address]" ?></td>
<td><?php echo"$row[book_name]" ?></td>
<td><?php echo"$row[feedback]" ?></td>
</tr>
<?php
 }
if ($i == 0) {
?>
<tr>
```

```
<td colspan="100%" class="text-center">No records</td>
</tr>
<?php }
?>
</tbody>
</table>
</div>
 </div>
</div>
</div>
</div>
<?php require_once '../include/footer.php'; ?>
</body></html>
```

- **Code for user module**

View book

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$user_id = $_SESSION['user_id'];
$query = "SELECT * FROM book AS b JOIN `category` AS c ON c.category_id=b.book_category
"."JOIN shop_keeper AS s ON b.shopkeeper_id=s.shop_keeper_rid WHERE is_active=1 AND
book_copy > 0";
$con = new DB();
$book = $con->executeSelect($query);
?>
<head>
<style type="text/css">
 table td { color: black; font-size:12pt; }
 .col-12 {
  background-color: skyblue; border: 1px solid black;
}
</style>
</head>
<br>
<div class="container-fluid " style="background-image:
url('../static/images/984c4e0369a2c.jpg');background-repeat: no-repeat;background-size:
cover;background-position: center;background-position: top;" >
 <div class="row justify-content-center  rounded">
<?php
if (!empty($book)) {
foreach ($book as $row) {
?>
```

```
<div class="card mb-3 mx-2 rounded" style="max-width: 650px;">
<div class="row g-0 align-items-center">
<div class="col-md-4">
<img src="../images/<?php echo "$row[photo]" ?> "
 alt="..."
 style="height: 60%;width: 100%;"
/>
</div>
<div class="col-md-8">
<div class="card-body">
<h5 class="card-title"><strong><?php echo "$row[book_name]" ?></strong></h5>
 <table class="table table-borderless text-dark">
 <tr>
<td><strong>Author Name</strong></td>
<td>:</td>
<td><?php echo "$row[author_name]" ?></td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>:</td>
<td><?php echo "$row[publisher_name]" ?></td>
 </tr>
 <tr>
 <td><strong>Category</strong></td>
 <td>:</td>
 <td><?php echo "$row[category_name]" ?></td>
 </tr>
<tr>
<td><strong>Description</strong></td>
 <td>:</td>
<td><?php echo "$row[description]" ?></td>
</tr>
<tr<td><strong>Price</strong></td>
<td>:</td>
<td>Rs.<?php echo "$row[book_price]" ?>/-</td>
</tr>
<tr>
 <td><strong>No.Of Copies</strong></td>
 <td>:</td>
 <td><?php echo "$row[book_copy]" ?></td>
 </tr>
 <tr>
 <td><strong>Shop Name</strong></td>
```

```
 <td>:</td>
 <td><?php echo "$row[shop_name]" ?></td>
 </tr>
<td>
<a href="order_book.php?id=<?php echo $row["book_rid"]; ?>&&cost=<?php echo
$row["book_price"]; ?>&&book_cpy=<?php echo $row["book_copy"]; ?>" class="btn btn-primary
btn-md px-4 mt-2">Order</a>
</td>
</table>
</div>
</div>
</div>
</div>
<?php
 }
 }
 ?>
</div>
</div>
<?php
require_once '../include/footer.php';
?>
```

Order book

```
<?php
require_once '../db/db.php';
require_once './home.php';
require_once '../include/header.php';
$book_id = $_GET['id'];
$book_cost = $_GET['cost'];
$book_cpy = $_GET['book_cpy'];
?>
<html>
<?php
?>
<body >
<div class="container">
<div class="row justify-content-center">
<div class="col-md-5">
<div class="card my-5">
<div class="card-body">
<h3 class="text-center">Order Book</h3>
<form action="u_action.php" method="post" id="formBookOrder" enctype="multipart/form-data">
<input type="hidden" name="command" value="orderBook"/>
```

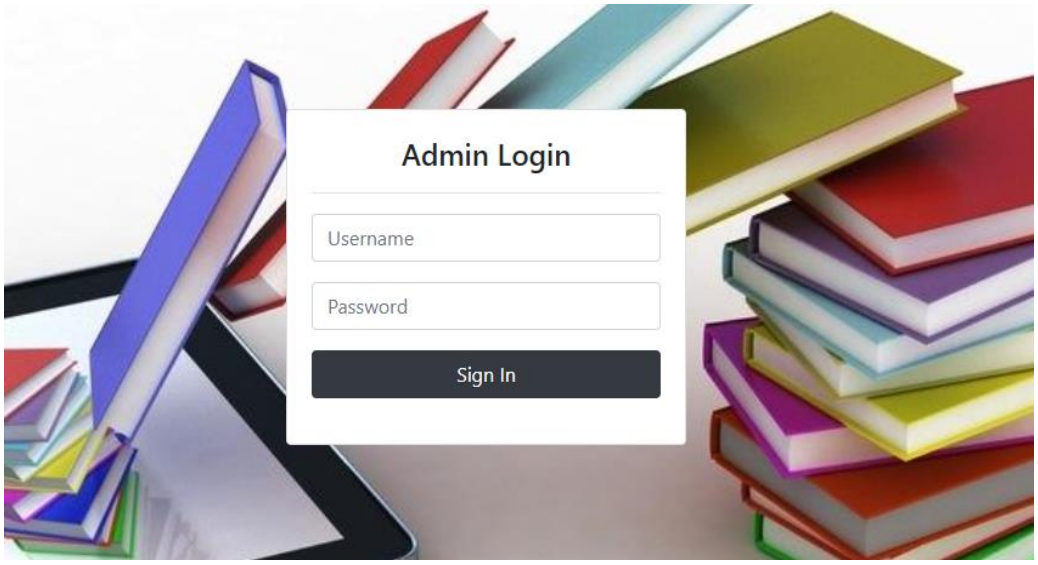
```
<input type="hidden" name="order_book_id"id="order_book_id"value="<?php echo $book_id;
?>"/>
<input type="hidden" name="book_cost" id="book_cost" value="<?php echo $book_cost; ?>"/>
<input type="hidden" name="book_cpy" id="book_cpy" value="<?php echo $book_cpy; ?>"/>
<input type="hidden" name="total_cost" id="total_cost"/>
<input type="hidden" name="orderId" id="orderId"/>
<div class="form-group mt-3">
<input class="form-control" value="Book Price (per copy) Rs.<?php echo $book_cost; ?>/-
"readonly="readonly"  autocomplete="off"/>
 </div>
 <div class="form-group my-3">
<input class="form-control d-none bg-light"  id="totalCost" name="totalCost" readonly="readonly"
autocomplete="off"/>
</div>
<div class="form-group my-3">
<input class="form-control " type="number" id="book_copies" name="book_copies" required
 placeholder="No. Of Book Copies" autocomplete="off"/>
</div>
<label id="totalCost" name="totalCost" class="d-none"></label>
<div class="form-group my-3">
<select class="form-control" name="payment_type" id="payment_type">
<option>--Choose Payment Type--</option>
<option value="upi">UPI</option>
<option value="cod">COD</option>
</select>
</div>
<div class="form-group">
 <button class="btn btn-dark btn-block" id="btnConfirmBookOrder">Confirm Order</button>
</div>
<div class="form-group">
<a id="btnPay" href="https://paytm.com/" onclick="pay_book()" target="_blank" class="btn btn-
danger btn-block d-none">Pay</a>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
<?php require_once '../include/footer.php'; ?>
<script src="../static/js/order.js" type="text/javascript"></script>
</body></html>
```

CHAPTER - 7

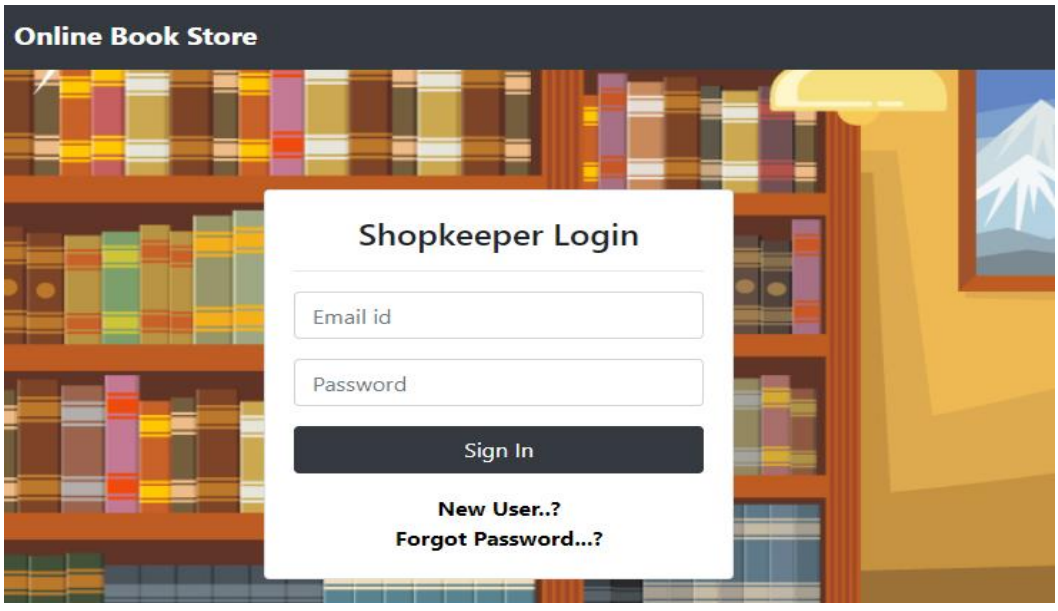
USER INTERFACE

7.1 Login

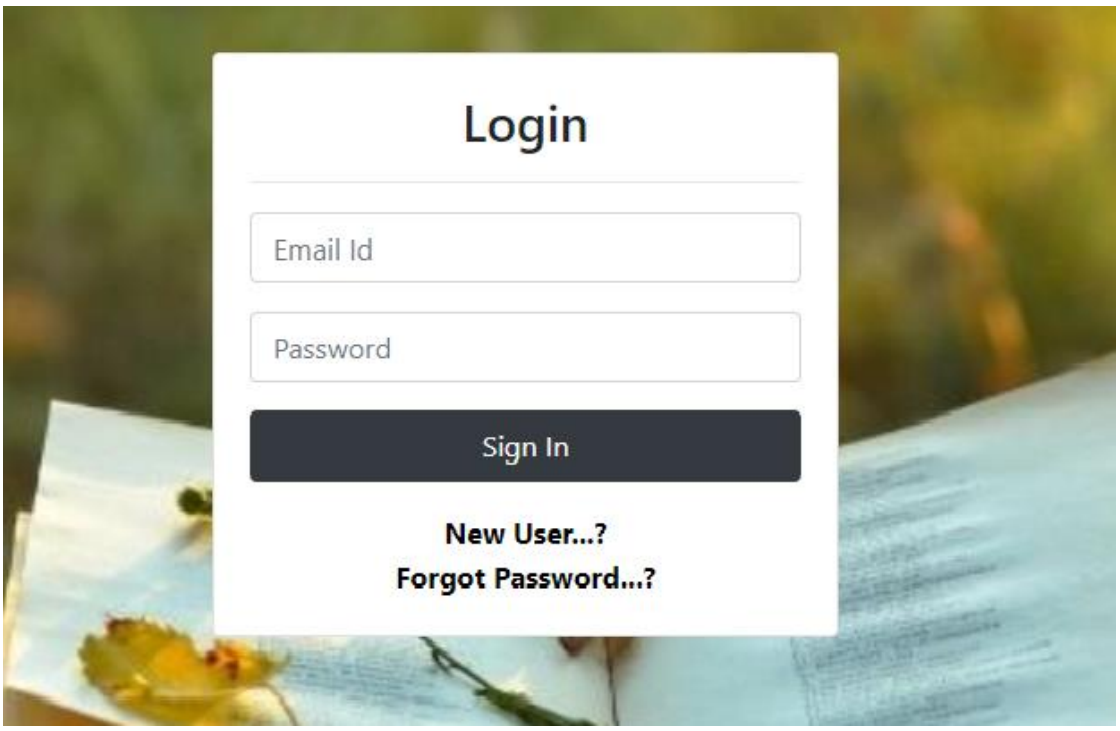
7.1.1 Admin Login



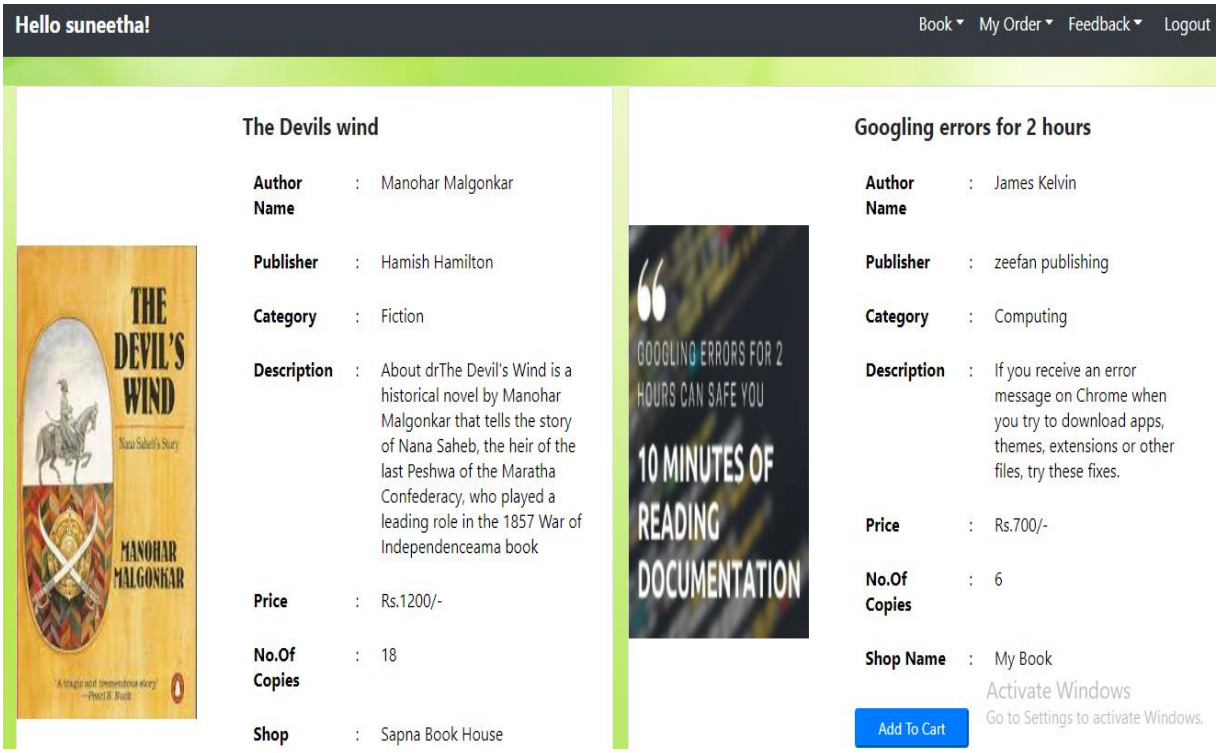
7.1.2 Shopkeeper Login



7.1.3 User Login

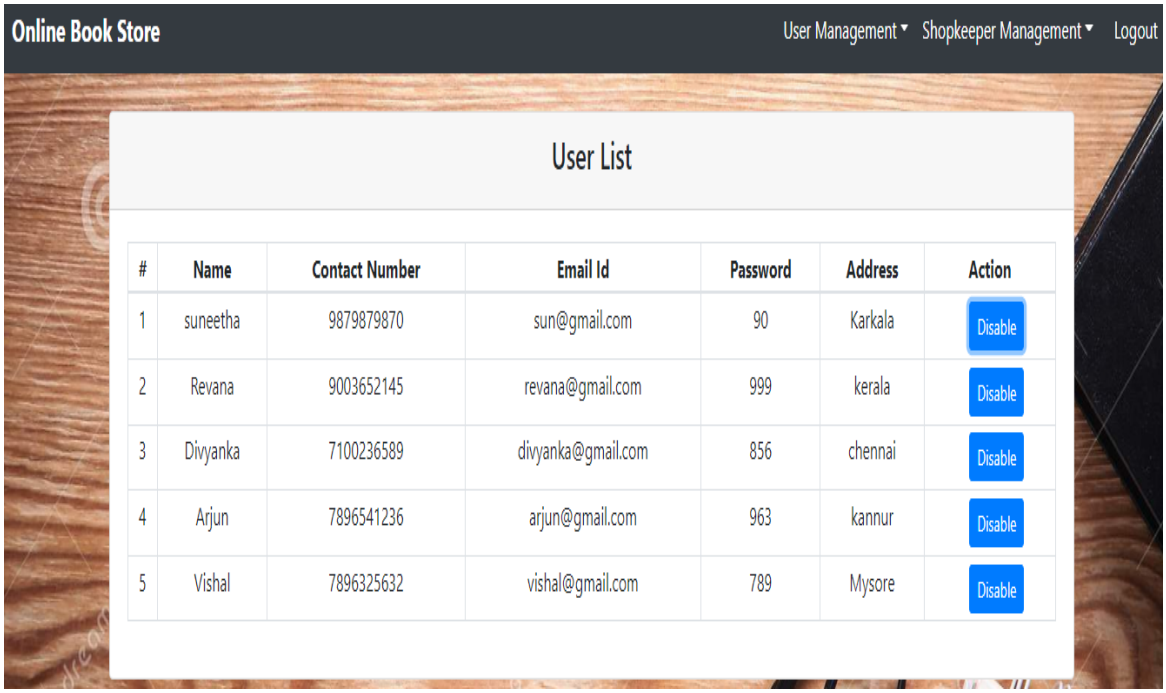


7.2 Main Screen

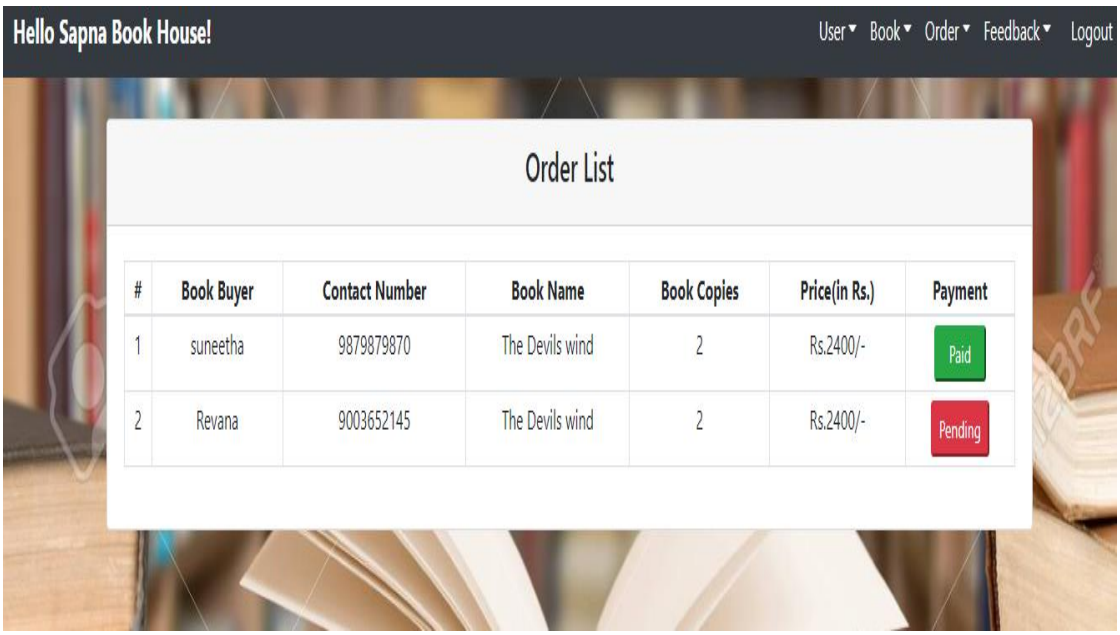


7.3 Menu

7.3.1 Menu for Admin



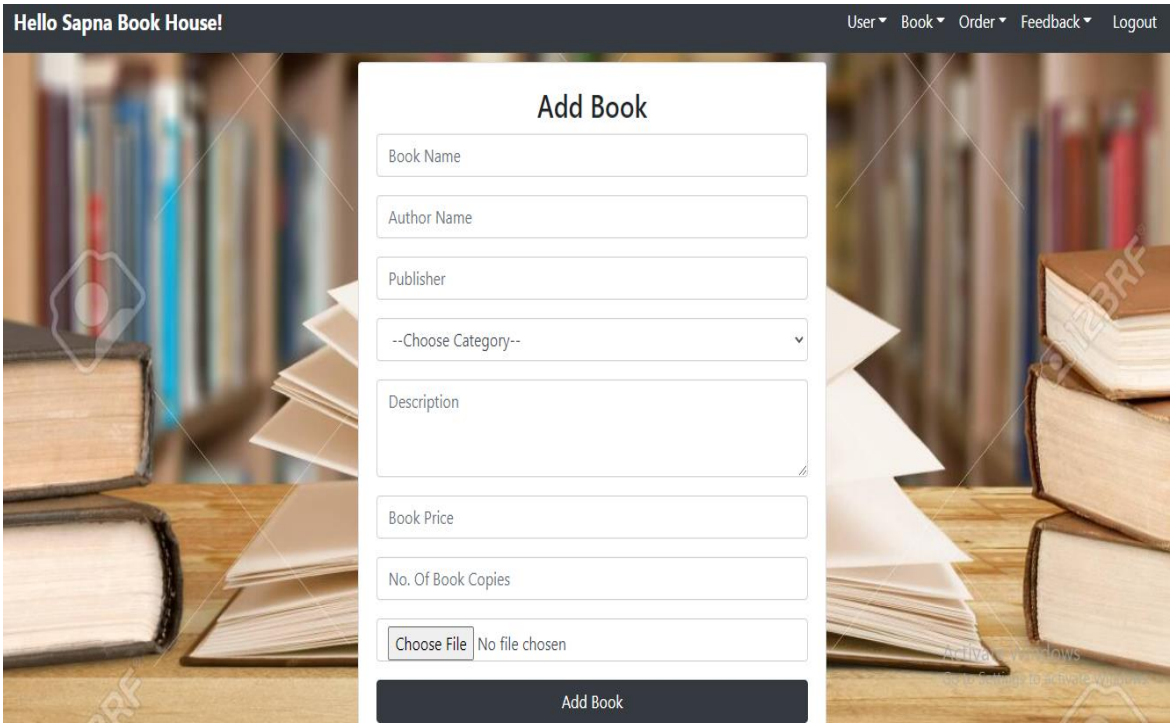
7.3.2 Menu for Shopkeeper



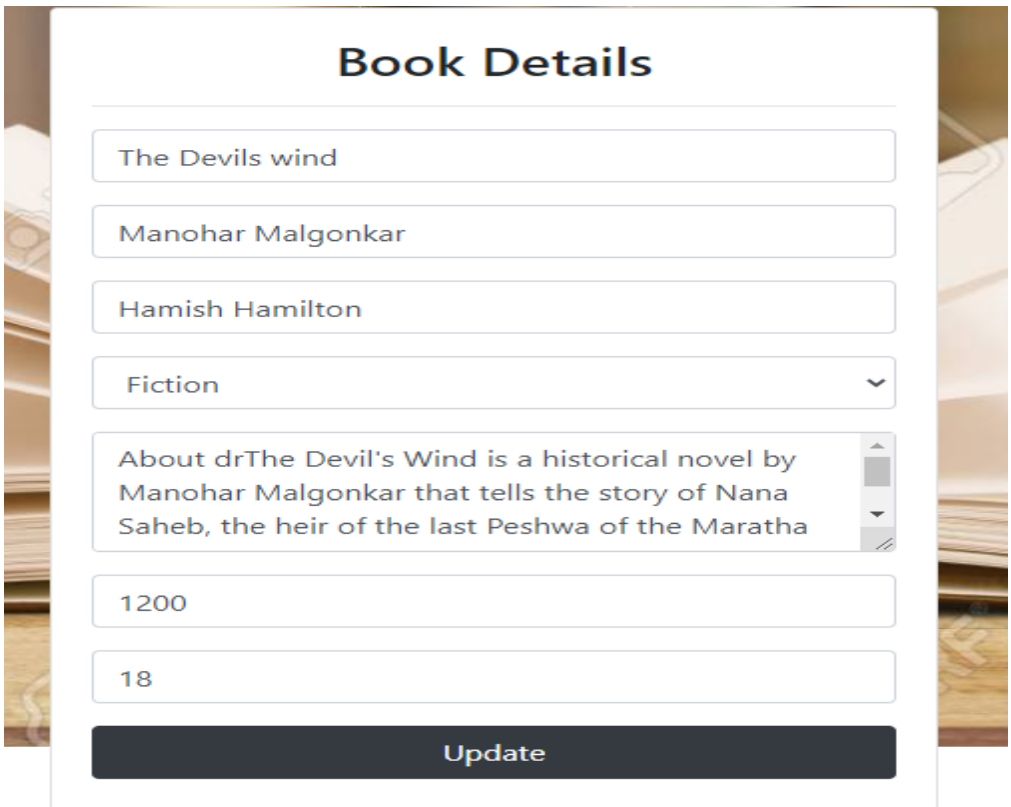
7.4 Data store/Retrieval/Update

7.4.1 For Shopkeeper

Add Book

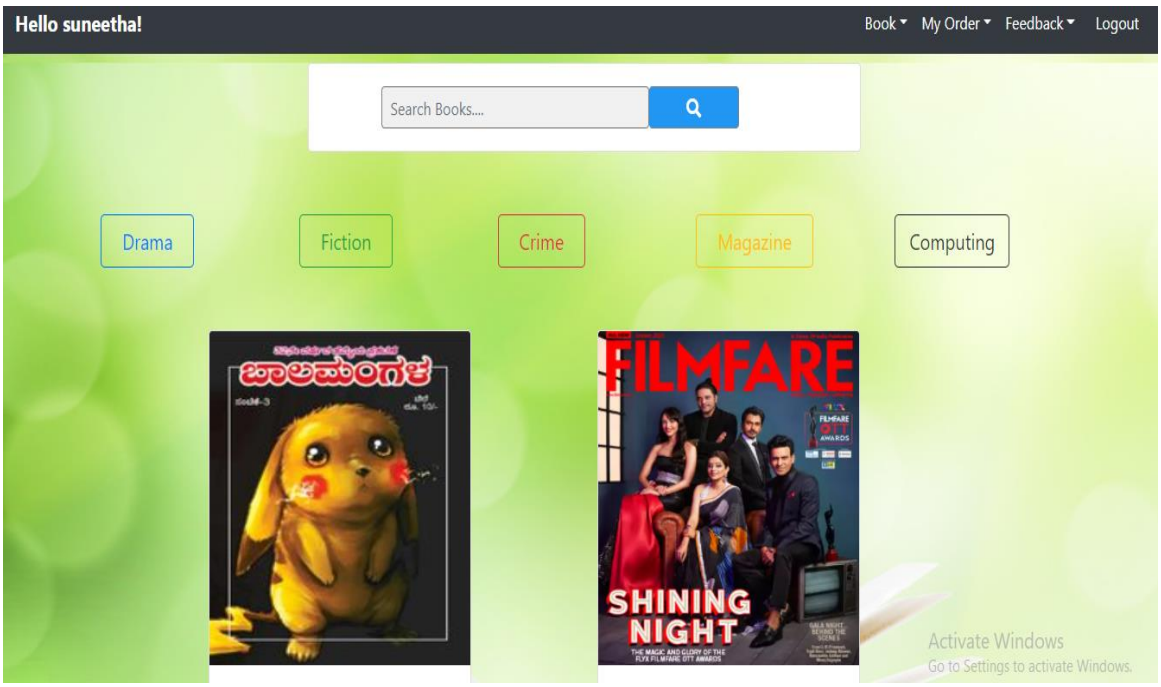


Update book

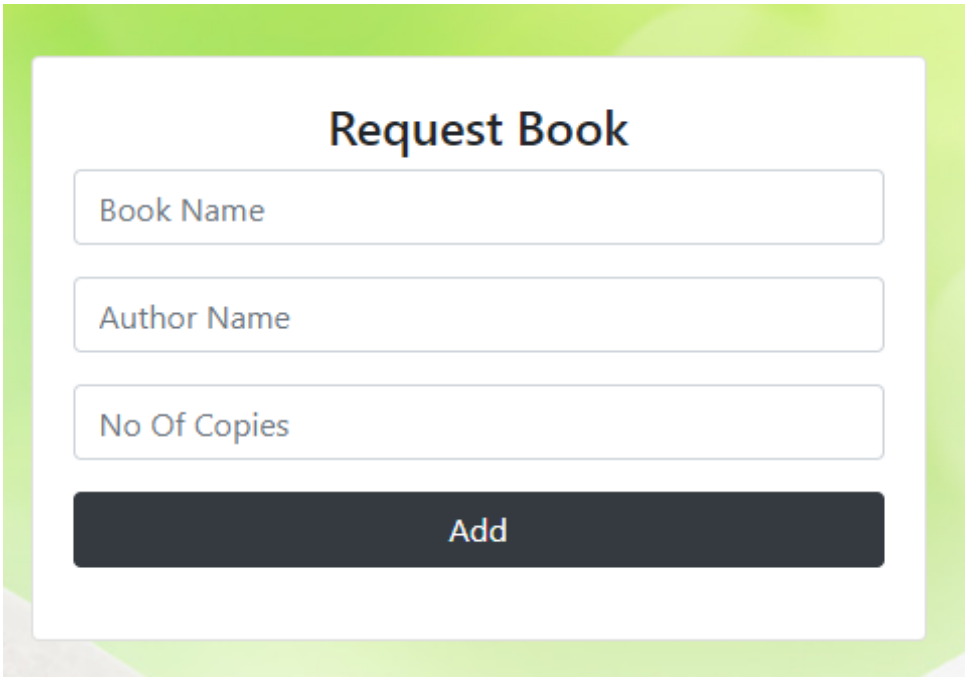


7.4.2 For User

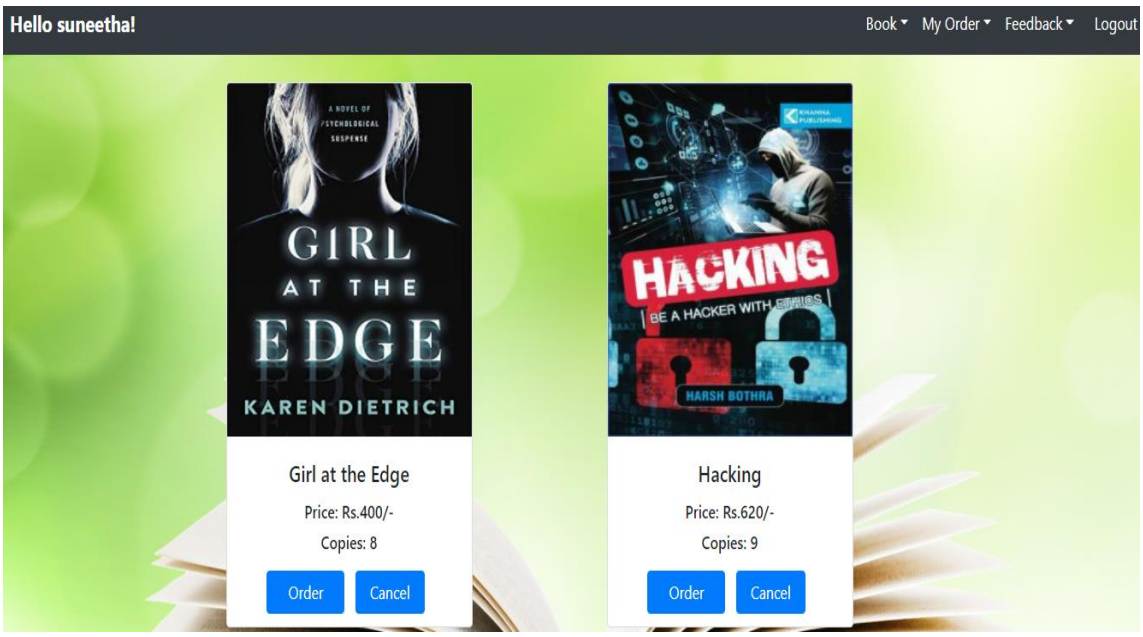
Search Book



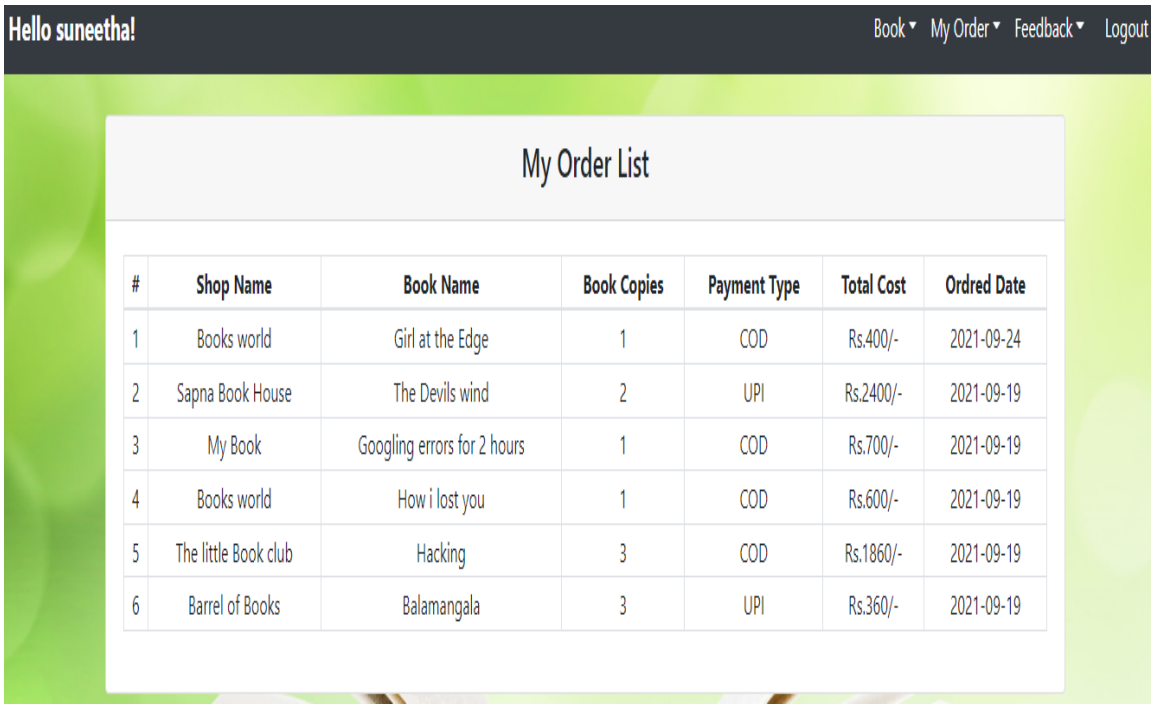
Request Book



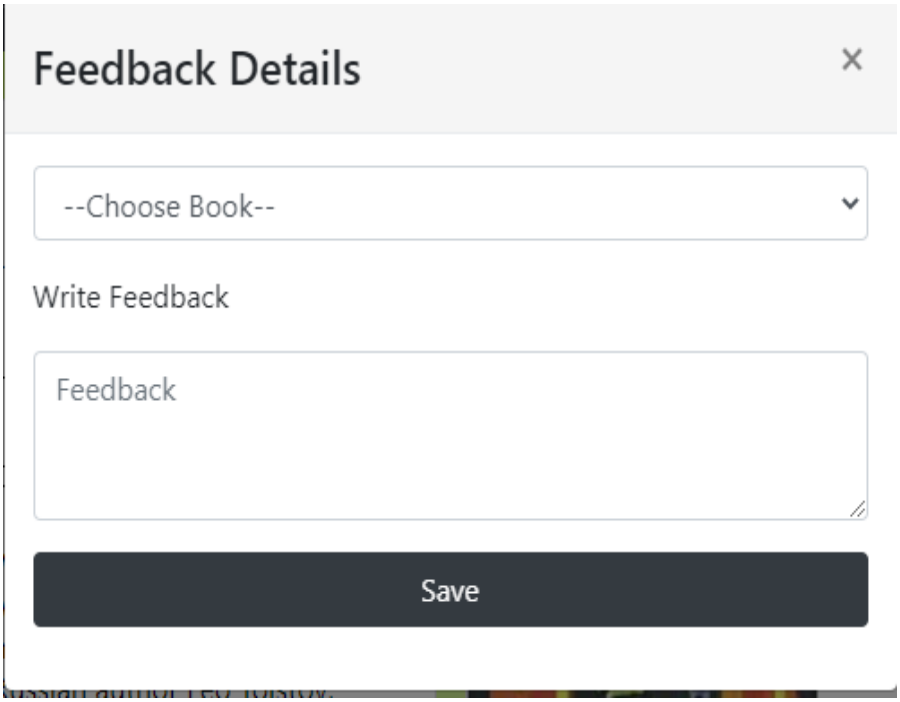
Order Books



View my orders

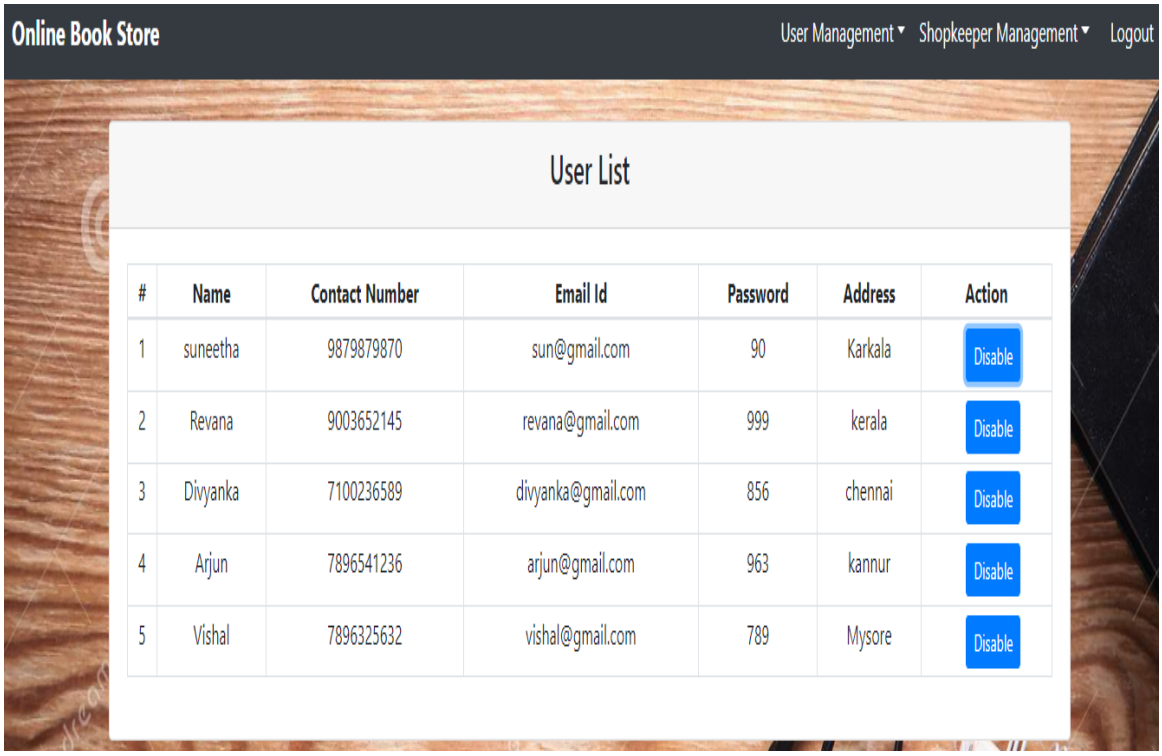


Feedback form

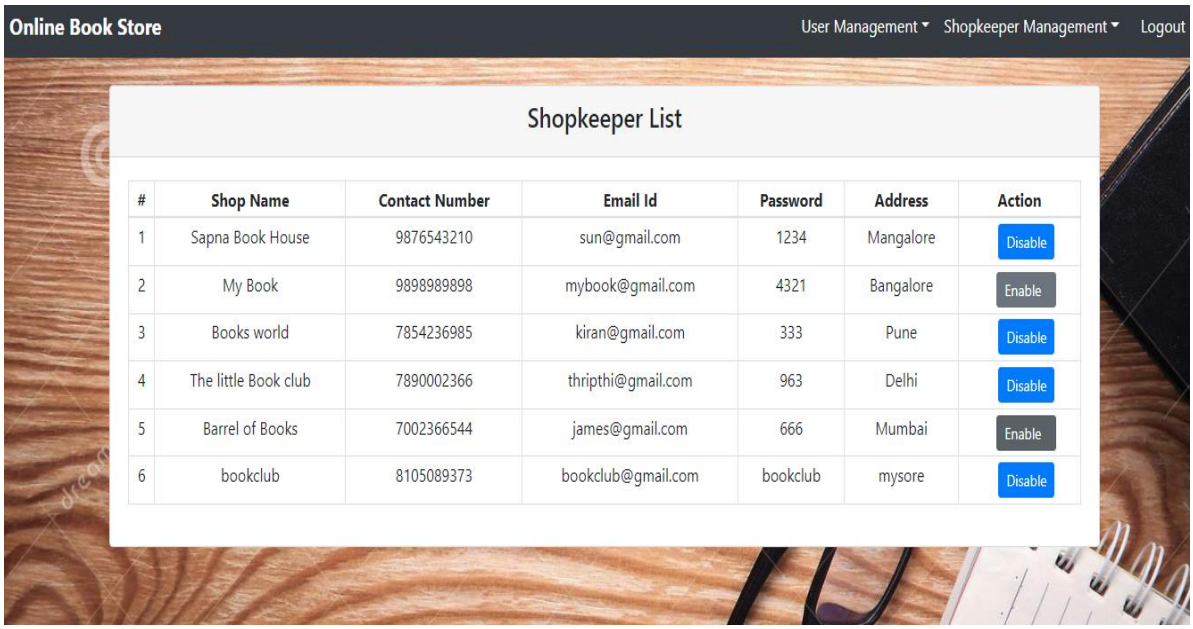


7.4.3 For Admin

User management

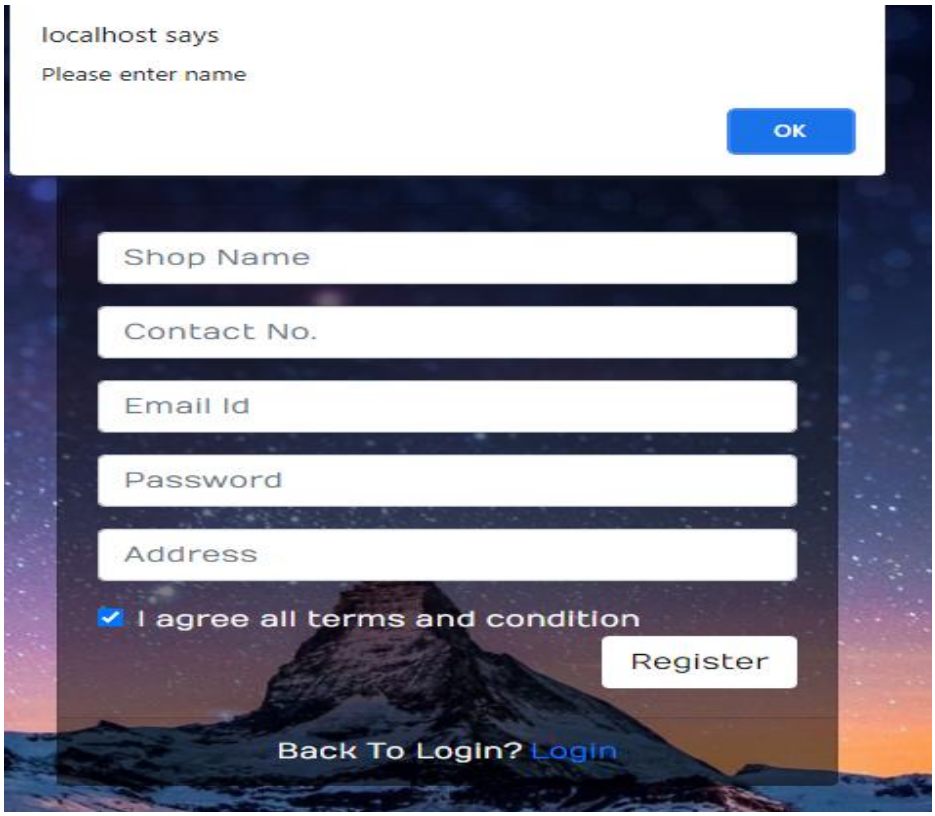


Shopkeeper Management

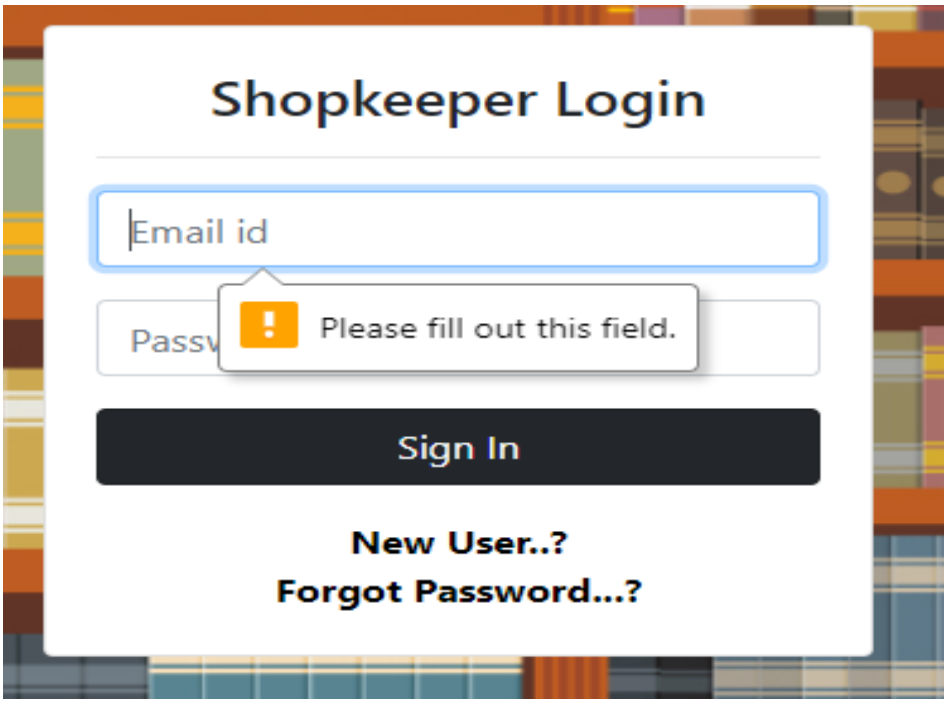


7.5 Validation

For Shopkeeper Registration

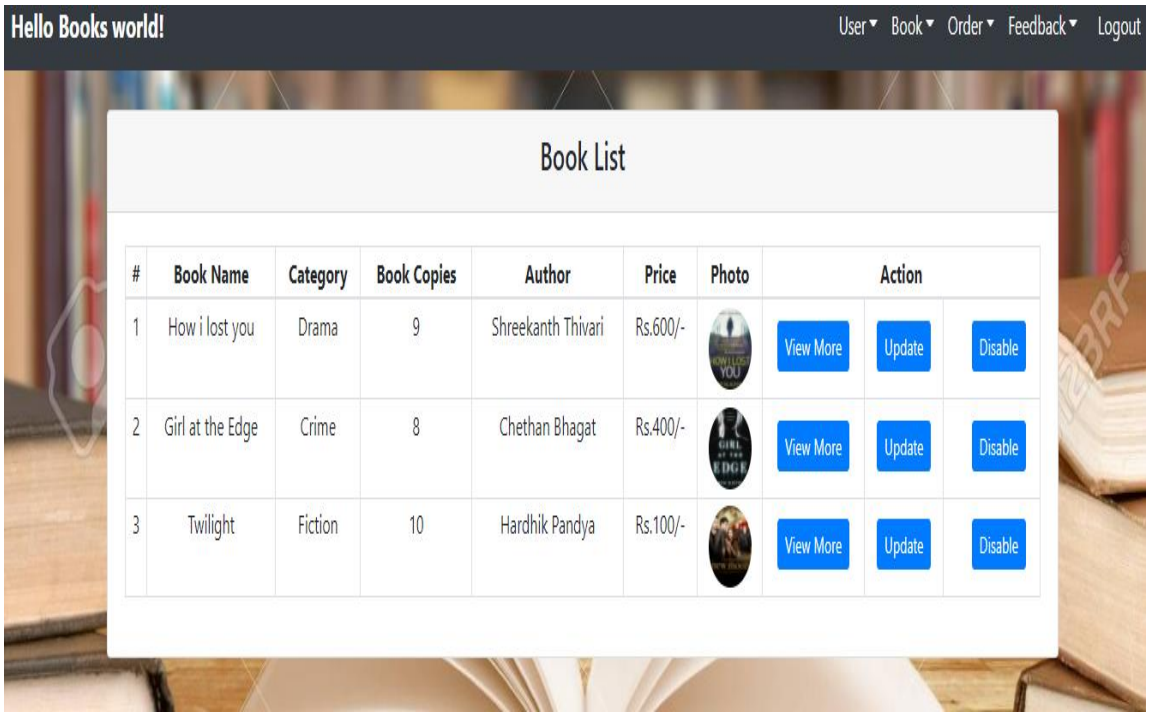


For Shopkeeper Login

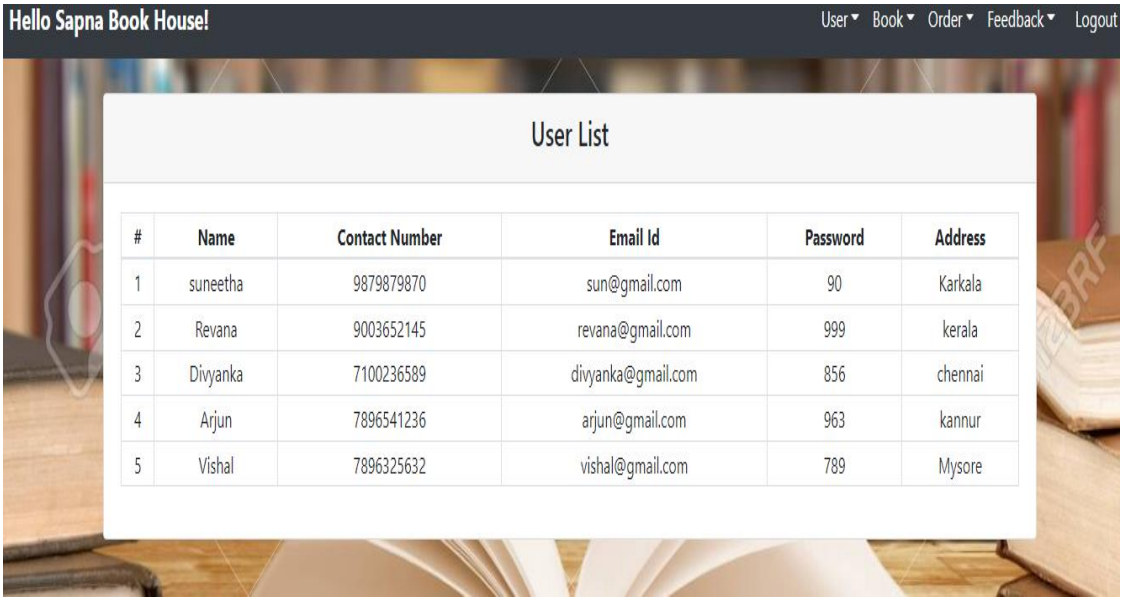


7.6 View

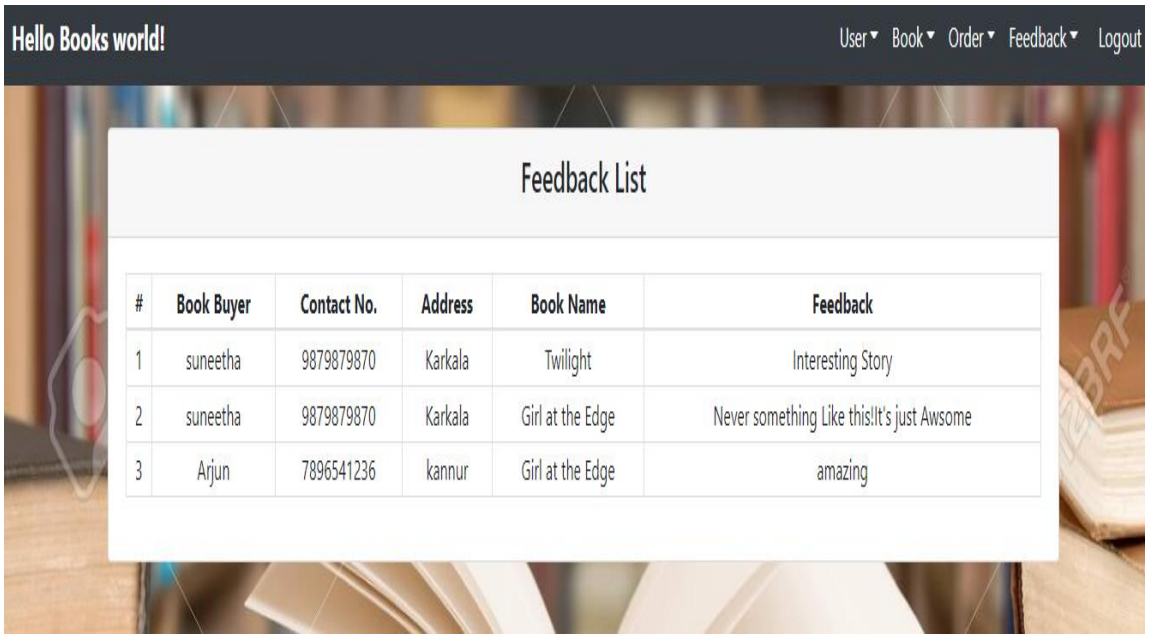
View Book



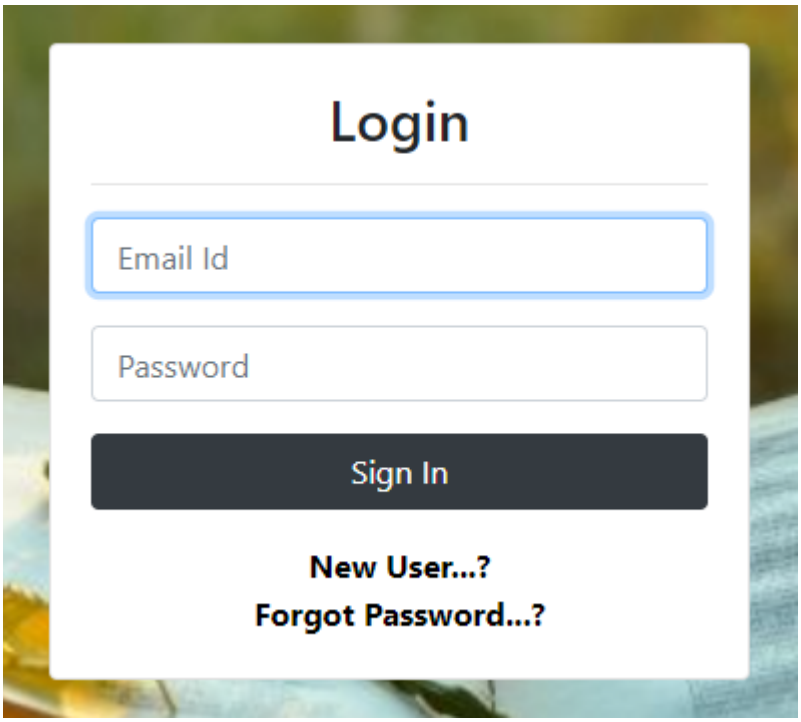
View users



View feedback



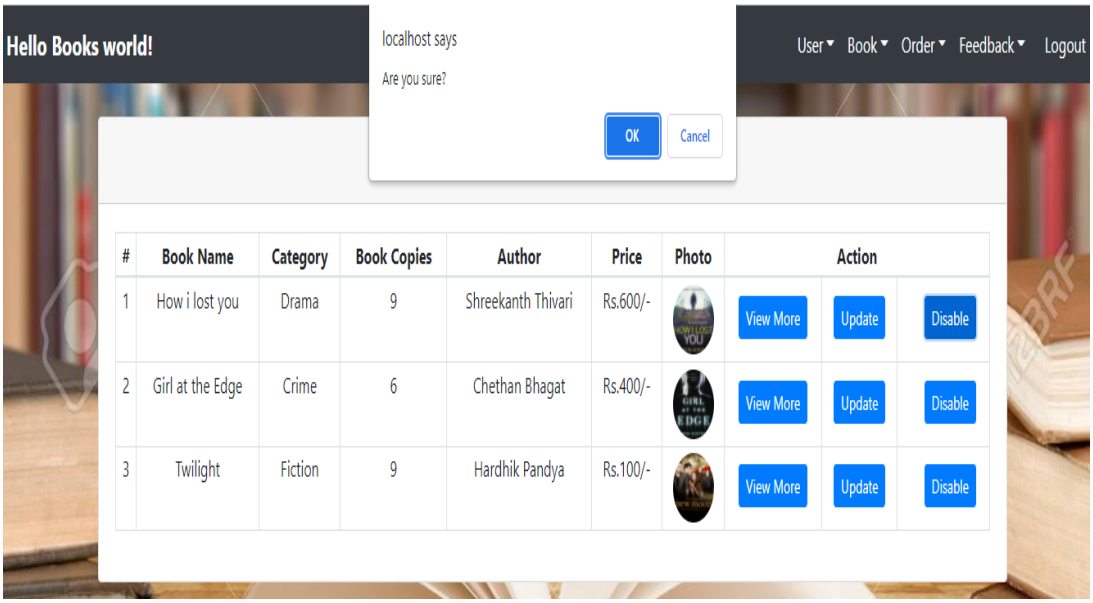
7.7 On Screen Report



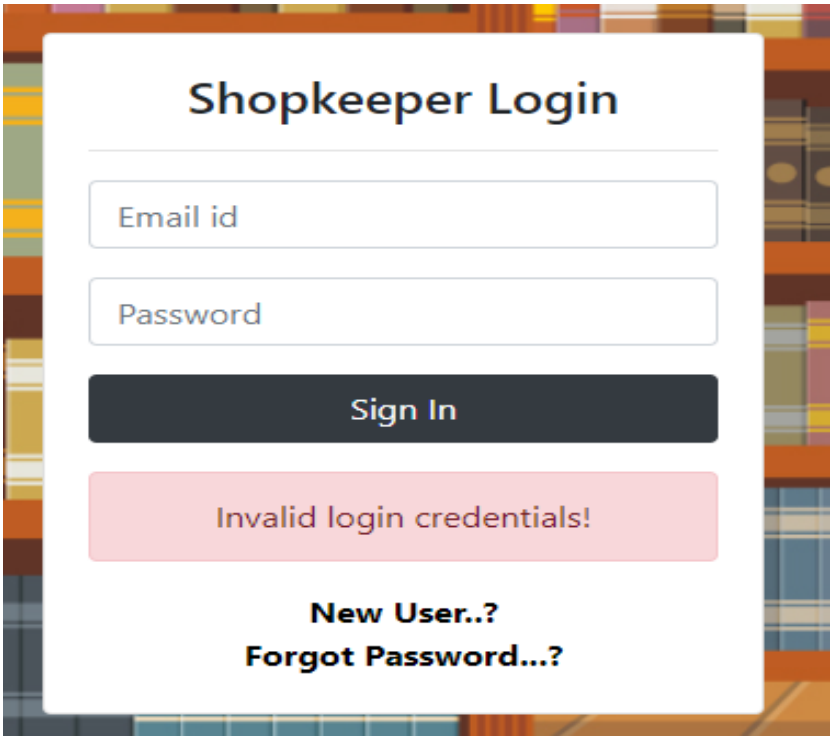
7.8 Data Report

	book_rid	book_name	author_name	publisher_name	book_price	photo	
	1	War And Peace	Leo Tolstoy	The Russian Messanger	900	1547386365612a46e7bd...	27B
	2	Malgudi Days	R.K Narayan	Indian Thought Publications	600	1750175854612a479a15...	27B
	5	ffdg	dfg	ffdgfdg	78	882080192612db861571...	26B
	6	xvgfg	fg	ffg	67	15842342186134b88526...	27B
	7	Hacking	Harsh Bothra	Khanna publishing	500	1798126285613e094573...	28B
	8	Blood Ice	Steeve james	Kevin publisher	600	12469300316l3e205032...	27B
	9	How I lost you	Adharsh Malhothra	Micaa publisher	330	528520170613e20baba0...	26B
	10	Stories we never Tell	Ricard	Ficaa	400	1019867535613e21197d...	28B
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	0K

7.9Alert message



7.10 Error message



CHAPTER – 8

TESTING

8.1 Introduction

Testing is a process of executing a program with the explicit intention of finding error. It is a process used to identify correctness, completeness and quality of developed computer software. There are many approaches to software testing, but effective testing of complex product is essential a process of investigating. Testing helps in verifying and validating if he software working as it is intended to work. This involves using static and dynamic methodologies to test the application. There are two methods for test case design.

Software testing has three main purposes:

- The verification process confirms that the software meet its technical specifications. A "specification" is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
- The validation process confirms that the software meets the business requirements.
- A defect is a variance between the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

There are two types of software testing:

- White Box Testing
- Black Box Testing

White Box testing:

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clear box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc. In order to implement white box testing the tester has to deal with the code and hence it is required possess knowledge of the coding and logic i.e. Internal working of the code.

Black Box testing

Black box testing takes the internal perspective of the test object to derived test cases. These tests can be functional or non-functional though usually functional. The test designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object's

internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance.

8.2 Testing Methodologies

- Unit Testing
- Integration Testing
- System Testing

8.2.1 Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage producers, are tested to determine ifthey are fit to use. Intuitively, one can view a unit as the smallest testable part of an application, In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object oriented programming a unit is often an entire interface, such as class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

8.2.2 Integration Testing

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These design items, i.e. assemblages (or group of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components with in assemblages interact correctively, for example across producers call of procedures activation, and this is done after testing individual modules, i.e. unit testing.

8.2.3 System Testing

A system testing of software or hardware is testing conducted on a complete, integrated system to evaluate system's compliance with its specified requirements. System testing falls within the scope of black box testing, and such as, should require no knowledge of the inner design of the integrated software components that have successfully passed integration testing and also the software components itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between software units that are integrated together(called assemblages) or between any of the assemblages and the hardware, System is more limited type of testing, it seeks to detect defects both within the inter-assemblages and also within the system as whole.

8.3 Testing Reports

System testing is the stage of implementation. This is to check whether system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: online response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

8.4 Test case

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as 'for condition x your derived result is y'; where as other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software testing process.

8.4.1.1 Admin Module

Login Page

SI No	Test Condition	Expected Result	Result
1	If Admin clicks login button without entering username and password	Message should be displayed "Please fill out this field"	Successful
2	If username is blank but password is entered	Message should be displayed "Please enter a valid username"	Successful
3	If password is blank but username is entered	Message should be displayed "Please enter the password"	Successful
4	If username and password are incorrect	Message should be displayed "Invalid login credentials"	Successful
5	If username and password are correct	Login	Successful

Admin Homepage

SI No	Test Condition	Expected Result	Result
1	When the admin clicks on “User Management”	To Enable or Disable user menu Appears	Successful
2	When the admin clicks “shopkeeper Management”	To Enable or Disable shopkeeper Appears	Successful

8.4.1.2 Shopkeeper Module

Registration page

Sl No	Test Condition	Expected Result	Result
1	When Shopkeeper clicks on the Create new account	Displays registration form	Successful
2	If Shop name is empty	Message should be displayed “Please enter name”	Successful
3	If contact number is empty or invalid	Message should be displayed “Please enter valid contact number”	Successful
4	If email is empty or invalid	Message should be displayed “Please enter a valid email”	Successful
5	If password is empty	Message should be displayed “Please enter a valid Password”	Successful
6	If address is empty	Message should be displayed “Please enter address”	Successful
7	When Shopkeeper Enters Shop Name, Contact No, Email id, password, Addresss and clicks on register	Message should be displayed Registered Successfully	Successful

Login page

Sl No	Test Condition	Expected Result	Result
1	If shopkeeper clicks login button without entering username and password	Message should be displayed “Please fill out this field”	Successful
2	If username is blank but password is entered	Message should be displayed “Please enter a valid username”	Successful

3	If password is blank but username is entered	Message should be displayed "Please enter the password"	Successful
4	If username and password are incorrect	Message should be displayed "Invalid login credentials"	Successful
5	If username and password are correct	Login	Successful

Homepage

Sl No	Test Condition	Expected Result	Result
1	When the Shopkeeper clicks the "User" in the menu.	It displays the details of registered users.	Successful
2	When the Shopkeeper clicks the "Book" in the menu.	It displays a page to add books as well as to view the added Books.	Successful
3	When the Shopkeeper clicks the "Orders" in the menu.	It displays the details of ordered books by the user as well as the requested book by the user.	Successful
4	When the shopkeepers clicks the "Feedback" in the menu.	It displays the feedback given by the users	Successful

Add Books

Sl No	Test Condition	Expected Result	Result
1	If Book name is empty	Please enter a Book name	Successful
2	If Author name is empty	Please enter Book Author name	Successful
3	If publisher is empty	Please enter Book Publisher name	Successful
4	If Book category is not selected	Please choose Book category	Successful
5	If Book description is empty	Please enter book description	Successful
6	If Book price is not Entered	Please enter Book price	Successful

7	If no of book copies is not entered	Please enter no of book copies	Successful
8	If Book image is not uploaded	Please upload book image	Successful
9	If all the data are valid and add button is clicked	Message should be displayed ”Added successfully”	Successful

8.4.1.3 User Module

Registration page

Sl No	Test Condition	Expected Result	Result
1	When user clicks on the Create new account	Displays registration form	Successful
2	If user name is empty	Message should be displayed “Please enter name”	Successful
3	If contact number is empty or invalid	Message should be displayed “Please enter valid contact number”	Successful
4	If email is empty or invalid	Message should be displayed “Please enter a valid email”	Successful
5	If password is empty	Message should be displayed “Please enter a valid Password”	Successful
6	If address is empty	Message should be displayed “Please enter address”	Successful
7	When user Enters Name, Contact No, Email id, password, Addresss and clicks on register	Message should be displayed Registered Successfully	Successful

Login page

SI No	Test Condition	Expected Result	Result
1	If user clicks login button without entering username and password	Message should be displayed “Please fill out this field”	Successful
2	If username is blank but password is entered	Message should be displayed “Please enter a valid username”	Successful

3	If password is blank but username is entered	Message should be displayed "Please enter the password"	Successful
4	If username and password are incorrect	Message should be displayed "Invalid login credentials"	Successful
5	If username and password are correct	Login	Successful

Homepage

Sl No	Test Condition	Expected Result	Result
1	When the user clicks the "book" in the menu.	It displays the no books,allows to search books and also request the books.	Successful
2	When the user clicks the "My orders" in the menu.	It displays my cart from which the book is ordered and view to view the ordered books.	Successful
3	When the user clicks the "Feedback" in the menu.	It allows user to write feedback regarding book purchased.	Successful

Order Books

Sl No	Test Condition	Expected Result	Result
1	When the user clicks the "My Cart" in the menu item	It displays collections of book for a user to order.	Successful
2	When the users clicks the "view " in the menu item.	It displays collections of books that user has ordered	Successful
3	If no of book copies is empty	Message should be displayed "Please enter the no of book copies	Successful
4	When user oders book copies more than the limit	Message should be displayed "Required copies not available"	Successful
5	If payment type is not selected	Message should be displayed "Please choose Payment type"	Successful
6	If all the data are valid and "confirm order" is clicked	Message should be displayed "ordered successfully"	Successful

Request Books

Sl No	Test Condition	Expected Result	Result
1	When the users clicks the “view request ” in the menu item.	It displays a page to request book	Successful
2	If Book name is empty	Message should be displayed “Please enter Book name”	Successful
3	If Author name is empty	Message should be displayed “Please enter book Author name”	Successful
4	If no of book copies is empty	Message should be displayed “Please entered required copies”	Successful
5	If all the data are valid and “Add ” is clicked	Message should be displayed “Added successfully”	Successful

Add feedback

Sl No	Test Condition	Expected Result	Result
1	When the users clicks the “feedback ”in the menu item.	It displays a page to write feedback	Successful
2	If Book name is Empty.	Message should be displayed “Please choose Book”	Successful
3	If Feedback is empty	Message should be displayed “Please write feedback details”	Successful
4	If all the data are valid and “Save” is clicked	Message should be displayed “Feedback added successfully”	Successful

CHAPTER-9

LIMITATIONS AND FUTURE ENHANCEMENT

Limitations

We have tried our best to present the information effectively, yet there are some limitations in this project which are listed below:

- There is no return policy ,if in case if the ordered books are damaged.
- Once the Book is odered ,users are not allowed to cancel their order.
- Proposed application does not allow users to do payment using debit/credit card transactions.
- Delivery mechanism is not included in this system.

Future Enhancement

We have tried our best to present the information effectively, yet, there can be further enhancement to the application is listed below:

- This web application can be extended to support different types of payments modes that helps the user.
- An android application can be developed to help the users to access the system easily and effectively.
- Cancel booking for users after ordering.
- To include delivery mechanism for fastest delivery.
- Replacement Policy for users if in case the book is damaged.

CHAPTER- 10

CONCLUSION

Therefore The system has the benefits of easy access because it is be developed as a platform independent web application, so the shopkeeper can maintain contact with their users, which may be access anywhere . Through our website, users can browse titles anytime and anywhere, and simply click “buy” when they’re ready. The convenience and immediacy of this experience encourages more purchases, as they won’t need to leave the safety and comfort of their homes. Purchasing from online bookstore will give readers the benefit of having their precious books delivered to them directly at home. This lowers the challenges of having to physically get to your store and back. When you shop online, you save a lot of time which you would have wasted wandering from one store to another if you were to shop at the conventional bookstore. Beside from that, there is no fixed time for online shopping. You can shop anytime. This is a huge plus to the homebound people or in a situation where the weather is hostile outside.

ANNEXURE- 1

ABBRIVATIONS AND ACRONYMS

I/O	-	Input / Output.
OS	-	Operating System.
DFD	-	Data Flow Diagram.
CSS	-	Cascading style sheet.
ADMIN	-	The Administrator.
CFD	-	Context Flow Diagram.
PHP	-	Preprocessor Hypertext.
GUI	-	Graphical User interface.
RAM	-	Random Access Memory.
SQL	-	Structured Query Language.
HTML	-	Hyper Text Markup Language.
SRS	-	Software Requirement Specifications.
IEEE	-	Institute of Electrical and Electronic Engineering.
XAMPP	-	Cross-Platform(X), Apache(A), MySQL(M), PHP(P) and Perl(P).
JS	-	Java Script.
CPU	-	Central processing unit.

ANNEXURE – 2

BIBLIOGRAPHY/ REFERENCES

Books

- W. Jason Gilmore, Beginning PHP and MySql: From Novice to Professional, 4th Edition, Apress, 2010.
- Timothy Boronczyk, et al, Beginning PHP6, Apache, MySql Web Development, Wiley India Pvt Ltd, 2009.
- AN INTEGRATED APPROACH TO SOFTWARE ENGINEERING,
  Edition: 6
  Author: PankajJalote
  Publication: Narosa Publishing House.

Websites

- http://www.w3cshools.com
- http://www.phpbuddy.com
- http://www.phpeasystep.com
- http://www.mysql.com