

I.K.G PUNJAB TECHNICAL UNIVERSITY



Batch(2019-2023)

“Online Book Store Management System”

**Submitted for the partial fulfillment of the
requirements of the B.Tech. 8th semester**

Submitted by

Name -Varsha Priya

Roll No-1916858

Branch -Btech.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

- 1.1 Background
- 1.2 Objective
- 1.3 Purpose, Scope and Applicability
 - 1.3.1 Purpose
 - 1.3.2 Scope
- 1.4 Achievements
- 1.5 Organization of Report

CHAPTER 2: SERVEY OF TECHNOLOGY

CHAPTER 3: REQUIREMENTS AND ANALYSIS

- 3.1 Problem Definition
- 3.2 Requirement Specification
- 3.3 Software and Hardware Requirements
 - 3.3.1 Hardware Requirements
 - 3.3.2 Software Requirements
- 3.4 Preliminary Product Description
- 3.5 Conceptual Models
 - 3.6.1 Module Specifications
 - 3.6.2 Screen Views
 - 3.6.3 Data Flow Diagrams

CHAPTER 4: SYSTEM DESIGN

- 4.1 Basic Modules
- 4.2 Data Design
 - 4.2.1 Schema Design
 - 4.2.2 Data Integrity and Constraints
- 4.3 Procedural Design
 - 4.3.1 Logic Diagrams

4.4 User Interface Design

4.5 Security Issues

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

5.2 Coding Details and Code Efficiency

5.2.1 Coding

5.2.2 Code Efficiency

5.3 Testing Approach

5.3.1 Unit Testing

5.3.2 Integrated Testing

5.4 Modifications and Improvements

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

6.2 User Documentation

CHAPTER 7: CONCLUSION

7.1 Conclusion

7.2 Limitations of the System

7.3 Future scope of the Project

CHAPTER 8: BIBLIOGRAPHY

Introduction:

Online shopping is the process whereby consumers directly buy goods, services etc. from a seller interactively in real-time without an intermediary service over the internet.

Online shopping is the process of buying goods and services from merchants who sell on the Internet. Since the emergence of the World Wide Web, merchants have sought to sell their books to people who surf the Internet. Shoppers can visit web stores from the comfort of their homes and shop as they sit in front of the computer. Consumers buy a variety of items from online stores. In fact, people can purchase just about anything from companies that provide their books online. Books, clothing, household appliances, toys, hardware, software, and health insurance are just some of the hundreds of books consumers can buy from an online store.

Many people choose to conduct shopping online because of the convenience. For example, when a person shops at a brick-and-mortar store, she has to drive to the store, find a parking place, and walk throughout the store until she locates the books she needs. After finding the items she wants to purchase, she may often need to stand in long lines at the cash register.

1.1 Background

Online shopping allows you to browse through endless possibilities, and even offers merchandise that's unavailable in stores. If you're searching for a niche product that may not be distributed locally, you're sure to find what you're looking for on the internet. What's even more useful is the ability to compare items, similar or not, online. You can search through multiple stores at the same time, comparing material quality, sizes and pricing simultaneously.

Say 'goodbye' to the days when you stood in line waiting, and waiting, and waiting some more for a store clerk to finally check out your items. Online shopping transactions occur instantly-saving you time to get your other errands done! Additionally, unlike a store, online shopping has friendly customer service representatives available 24 hours a day, 7 days a week to assist you with locating, purchasing and shipping your merchandise.

1.2 Objective

My objective is to design such an application using which one can say 'goodbye' to the days when you stood in line waiting, and waiting some more for a store clerk to finally

check out your items. Online shopping transactions occur instantly-saving you time to get your other errands done! Additionally, unlike a store, online shopping has friendly customer service representatives available 24 hours a day, 7 days a week to assist you with locating, purchasing and shipping your merchandise.

My main aim is to design such a book store where customer can visit our site anytime of the day from anywhere to view the available books, choose any of them and can order by paying online or can opt for cash on delivery as well. The administrator will regularly add any new books available to them for sale. The administrator will take books from the reputed publishers and vendors only.

1.3 Purpose and Scope

1.3.1 Purpose

Online Shopping system would have the following goals.

- Provide a web user interface to add, view, delete records in different areas.
- Provide a user interface to enter computer details.
- Provide a user interface to change details of all the computers and accessories.
- Provide a user interface for users to explore the store and choose items to buy.

1.3.2 Scope

The main scope and deliverables of the project would be to:

- Understand and prepare detailed requirement and specifications
- Prepare high level and detailed design specifications of the system
- Prepare Test Plan and Test cases
- Develop the system and coding
- Perform unit testing, integration and system testing
- Demonstrate a bug free application after suitable modification if needed.

1.4 Achievements

- By successfully implementing the project, a substantial knowledge has been acquired on the implementation of a database system using .net technologies.

This knowledge will be useful in the future in creating any type of desktop application or online database systems.

2. SURVEY OF TECHNOLOGY

In a desktop application like Laboratory Management System, there is a scope for a large number of platforms, languages and frameworks to choose from. Before selecting from this large array of technologies, the following aspects, which are characteristic to windows based application like this one, have been kept in mind:

- Data validation
- Performance
- Reliability
- Scalability
- Security
- Portability
- Performance
- Time constraint
- Cost constraint

The various technologies available for consideration are as follows:

Operating System: Windows 7

Client Side Scripting:

- HTML
- CSS
- JavaScript

Server Side Scripting: PHP

Database Tool: My SQL

Testing Server: Apache

Other Software Used:

- Adobe Dreamweaver
- Adobe Photoshop
- Wamp Server

HTML

HTML or **HyperText Markup Language** is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example . The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

CSS

CSS was first developed in 1997, as a way for Web developers to define the look and feel of their Web pages. It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on the markup of content, without worry about the design and layout.

CSS didn't gain in popularity until around 2000, when Web browsers began using more than the basic font and color aspects of CSS.

Web Designers that don't use CSS for their design and development of Web sites are rapidly becoming a thing of the past. And it is arguably as important to understand CSS as it is to know HTML - and some would say it was more important to know CSS.

Style sheet refers to the document itself. Style sheets have been used for document design for years. They are the technical specifications for a layout, whether print or online. Print designers use style sheets to insure that their designs are printed exactly to specifications. A style sheet for a Web page serves the same purpose, but with the added functionality of also telling the viewing engine (the Web browser) how to render the document being viewed.

PHP:

PHP (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions `<?php` and `?>` that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

MYSQL:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL Web site (<http://www.mysql.com/>) provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (<http://www.mysql.com/company/legal/licensing/>).

3. REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

Problem Definition and Need for the New System

- Online Book Store is a specific requirement of the client that integrates the buying and selling services specifically to their customers.
- Reports can be generated at any time within few seconds, so that manual labor is not required, and also analysis can be performed much more frequently which helps in taking decision.
- The details regarding all users, books can also be maintained as their information is very helpful and sometimes becomes a critical requirement.
- Allows user to get registered from their places and transact for the required product.
- To overcome these problems we develop “**Online Book Store**”.

SYSTEM REQUIREMENTS SPECIFICATIONS

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The software specification document satisfies the following:-

- It specifies the external system behaviours.
- It specifies constraints on the implementation.
- It is easy to change.
- It serves as reference tool for system maintainers.
- It record forethought about the life cycle of the system.
- It characterizes acceptable response to undesired events.

User Class and Characteristics:

- General public
- Customers
- Administrator
- General public can use the system to see the books, their prices and quantity available.
- Non registered user cannot buy the books.
- Customers are using for viewing and buying the books.
- Customer can also write feedbacks for books and services
- Administrators can add, edit & delete books and provide services to the customer.
- Administrator can see the daily sell. Can also see the feedback given by the customer.
- Administrator maintaining the deliveries.

Functional Requirements:

- The System must provide following functionalities—
- Keeping records of registration of customers.
- Keeping the records of books.
- Keeping the daily sell.
- Storing the feedback given by the customer.
- Keeping details about the product it is delivered or not. etc.
- Storing the items selected by the customer in the temporary storage.

Non Functional Requirements:

Following Non-functional requirements will be there in the online shopping portal.

- Secure access of confidential data (customer's details).
- 24 X 7 availability.
- Better component design to get better performance at peak time.

Flexible service based architecture will be highly desirable for future extension Non functional requirements define system properties and constraints It arise through user

needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on.

Various other Non-functional requirements are:

1. Security
2. Reliability
3. Maintainability
4. Portability
5. Extensibility
6. Reusability
7. Application Affinity/Compatibility
8. Resource Utilization

External Interface Requirements:

User Interface:

User of the system will be provided with the Graphical user interface, there is no command line interface for any functions of the product.

Hardware Interface:

Hardware requirements for running this project are as follows:

Processor: - Pentium I or above.

RAM: - 128 MB or above.

HD: - 20 GB or above.

Software Interface:-

Software required to make working of product is:-

Front end- HTML/PHP

Back end- My SQL

3.5 Conceptual Models

DATA FLOW DIAGRAM

What it is?

The Data Flow Diagram shows the flow of data or information. It can be partitioned into single processes or functions. Data Flow Diagrams can be grouped together or decomposed into multiple processes. There can be physical DFD's that represent the physical files and transactions, or they can be business DFD's (logical, or conceptual).

When it's used?

The DFD is an excellent communication tool for analysts to model processes and functional requirements. One of the primary tools of the structured analysis efforts of the 1970's it was developed and enhanced by the likes of Yourdon, McMenamin, Palmer, Gane and Sarson. It is still considered one of the best modeling techniques for eliciting and representing the processing requirements of a system.

Used effectively, it is a useful and easy to understand modeling tool. It has broad application and usability across most software development projects. It is easily integrated with data modeling, workflow modeling tools, and textual specs. Together with these, it provides analysts and developers with solid models and specs. Alone, however, it has limited usability. It is simple and easy to understand by users and can be easily extended and refined with further specification into a physical version for the design and development teams.

The different versions are Context Diagrams (Level 0), Partitioned Diagrams (single process only -- one level), functionally decomposed, leveled sets of Data Flow Diagrams.

Data Store

It is a repository of information. In the physical model, this represents a file, table, etc. In the logical model, a data store is an object or entity.

DataFlows

DFDs show the flow of data from external entities into the system, showed how the data moved from one process to another, as well as its logical storage. There are only four symbols:

- ❑ Squares representing **external entities**, which are sources or destinations of data.
- ❑ Rounded rectangles representing **processes**, which take data as input, do something to it, and output it.
- ❑ Arrows representing the **data flows**, which can either, be electronic data or physical items.
- ❑ Open-ended rectangles representing **data stores**, including electronic stores such as databases or XML files and physical stores such as or filing cabinets or stacks of paper.

There are several common modeling rules for creating DFDs:

- ❑ All processes must have at least one data flow in and one data flow out.
- ❑ All processes should modify the incoming data, producing new forms of outgoing data.
- ❑ Each data store must be involved with at least one data flow.
- ❑ Each external entity must be involved with at least one data flow.
- ❑ A data flow must be attached to at least one process.

DFDs are nothing more than a network of related system functions and indicate from where information is received and to where it is sent. It is the starting point in the system that decomposes the requirement specifications down to the lowest level detail.

The four symbols in DFD, each of which has its meaning. They are given below:

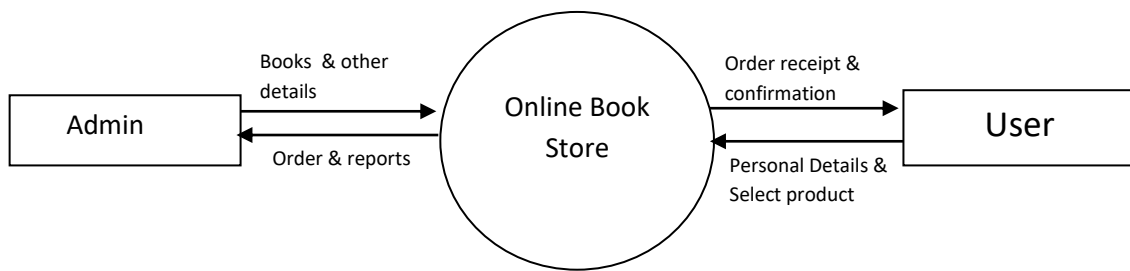
- ❑ External entities are outside to system but they either supply input data in the system or use the system output. These are represented by square of rectangle. External entities that supply data into a system are sometimes called Sources. External entities that use system data are sometimes called sinks.

- ❑ Dataflow models the passages of data in the system and are represented by line by joining system components. An arrow indicates the direction of the flow and the line is labeled by the name of the dataflow.
- ❑ Process shows what the systems do. Each process has one or more data inputs and one or more data outputs. Circles in DFD represent them. Each high level process may be consisting of more than one lower level processes. Process will be expanded in subsequent level DFD. A circle or a bubble represents a process that transforms incoming data flow into outgoing dataflow.

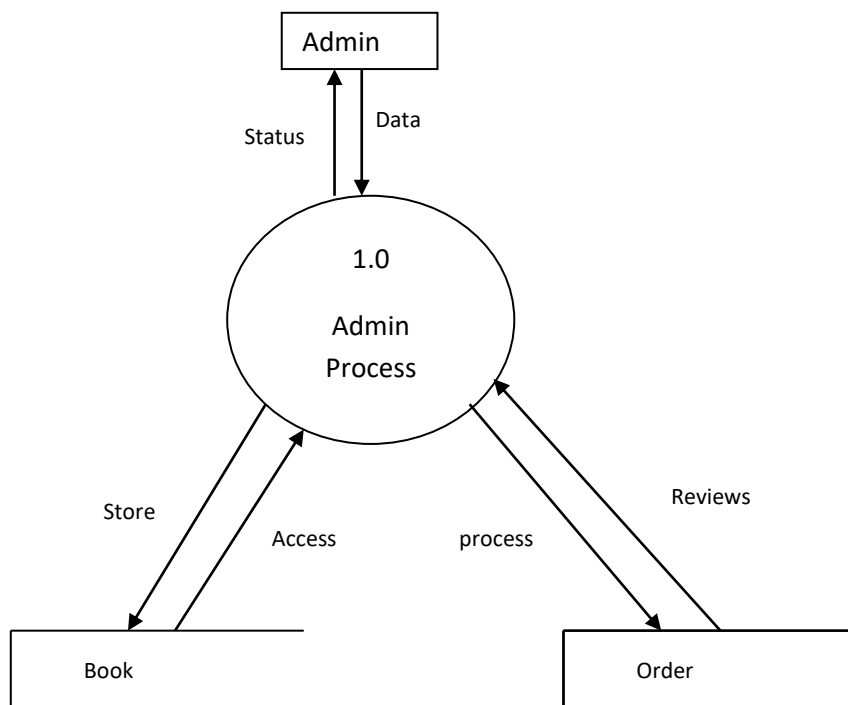
The high level processes in a system are:

- ❑ Receivable process.
 - ❑ Verifiable process.
 - ❑ Disposal process.
- ❑ File or data store is a repository of data. They contain data that is retained in the system. Process can enter data into data store or retrieve data from the data store. An open rectangle is a data store, data at rest.

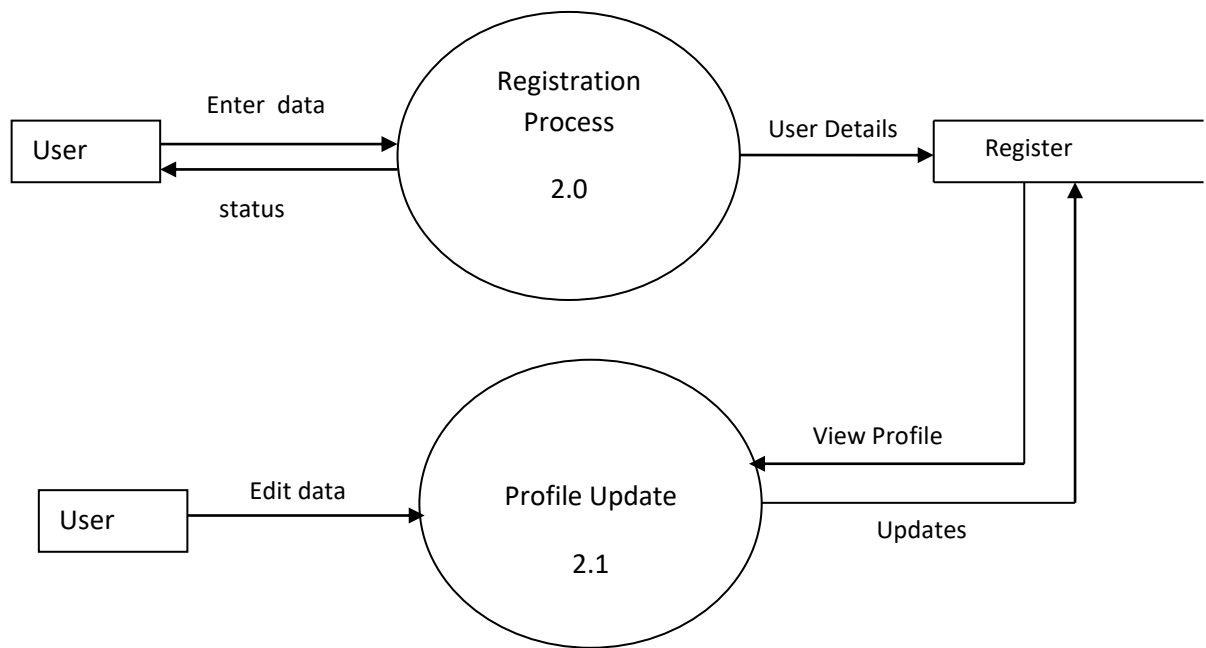
0-Level DFD:



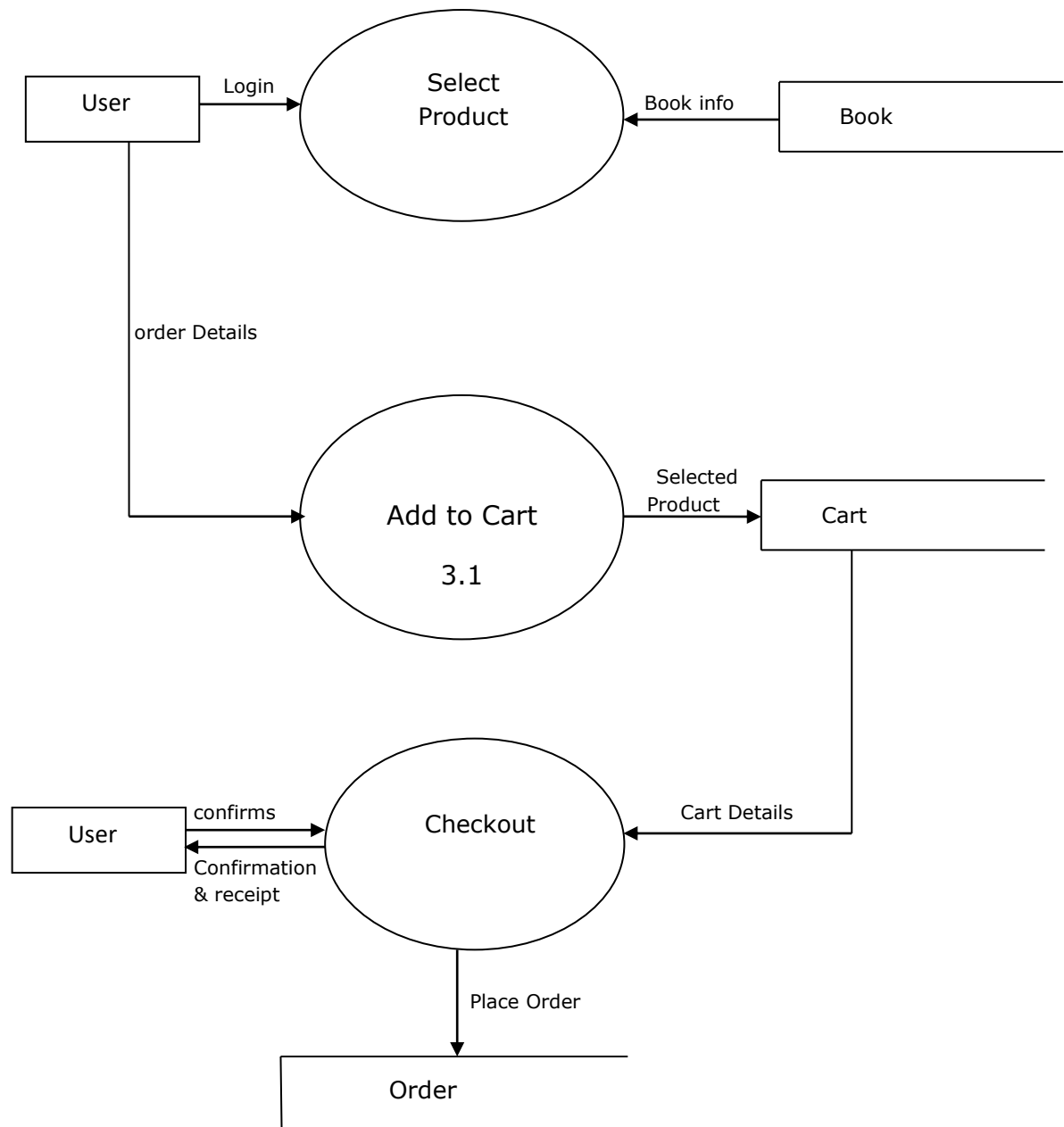
CONTEXT DTAGRAM



DFD for Admin Process



DFD For User Registration and Profile Update



DFD for shopping and checkout process

Entity-Relationship Model

Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database.

Basic Constructs of E-R Modeling

The ER model views the real world as a construct of entities and association between entities.

Entities

Entities are the principal data object about which information is to be collected. Entities are classified as independent or dependent (in some methodologies, the terms used are strong and weak, respectively). An independent entity is one that does not rely on another for identification. A dependent entity is one that relies on another for identification. .

Relationships

A Relationship represents an association between two or more entities. Relationships are classified in terms of degree, connectivity, cardinality, and existence.

Attributes

Attributes describe the entity of which they are associated. A particular instance of an attribute is a value. The domain of an attribute is the collection of all possible values an attribute can have. The domain of Name is a character string.

Classifying Relationships

Relationships are classified by their degree, connectivity, cardinality, direction, type, and existence. Not all modeling methodologies use all these classifications.

Degree of a Relationship

The degree of a relationship is the number of entities associated with the relationship. The n-ary relationship is the general form for degree n. Special cases are the binary, and ternary, where the degree is 2 and 3 respectively.

Connectivity and Cardinality

The connectivity of a relationship describes the mapping of associated entity instances in the relationship. The values of connectivity are "one" or "many". The cardinality of a relationship is the actual number of related occurrences for each of the two entities. The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many.

Direction

The direction of a relationship indicates the originating entity of a binary relationship. The entity from which a relationship originates is the parent entity; the entity where the relationship terminates is the child entity.

The direction of a relationship is determined by its connectivity type. An identifying relationship is one in which one of the child entities is also a dependent entity. A non-identifying relationship is one in which both entities are independent.

Existence

Existence denotes whether the existence of an entity instance is dependent upon the existence of another, related, entity instance. The existence of an entity in a relationship is defined as either mandatory or optional.

Generalization Hierarchies

A generalization hierarchy is a form of abstraction that specifies that two or more entities that share common attributes can be generalized into a higher level entity type called a supertype or generic entity. The lower-level of entities become the subtype, or categories, to the supertype. Subtypes are dependent entities.

ER Notation

The symbols used for the basic ER constructs are:

- ❑ Entities are represented by labeled rectangles. The label is the name of the entity.
- ❑ Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- ❑ Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- ❑ Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- ❑ Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.
- ❑ Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

SYSTEM DESIGN

Introduction:

System design is the solution of a “how to approach to the creation of the new system. It is composed of several steps. It facilitates the understanding and provides the procedural details necessary for implementation of the system recommended in the feasibility study. Emphasis is given on translating the performance requirements into design specification. Design goes through logical and physical stages of development.

Logical design reviews the present physical system; prepares input and output specification; make editing; security and control specification; details the implementation plan, and prepare logical design walk through. The physical design maps out the details of the physical system; plans the system implementation plan and specifies hardware and software. System design translates the system requirement into the ways of the system as recommended in the feasibility study. Thus the system design is the translation from

user-oriented document to a programmer or a database personal oriented document. System design is a highly creative process that can be greatly facilitated by the following:-

- Strong Problem Definition
- Pictorial description of the Existing System
- Set of Requirements of the new system

Modules Description:

1. Registration: Customer can register their account here to continue shopping.
2. Admin: Admin can add books, check orders and make sure the orders are delivered on time and can confirm payments by the customers.
3. Shopping Cart: Customers after login can browse through the different books and choose one or more products and can add them to cart.
4. Payment: Cash on Delivery facility is available.

3.2 INPUT DESIGN

Very careful attention had to be given to input design, which is a major part of the overall system design. In order to make the data entry as easy, logical and error free as possible, specific standards had been followed. Validation checks, provided in the system prevented the user in entering incorrect, erroneous data. This made sure that, only valid data had been available for data processing. If valid data was entered, then meaningful error messages had been prompted to enter correct data. The interactive screen formats facilitate the entry of valid data.

3.2.1 VALIDATIONS:

Some fields are having only number, as an I/P. For this key ASCII is checked. If they entered characters, it would display the message to enter number only. Exchange rates field will be validated for number and dot symbols.

3.2.2 INPUT DESIGN OBJECTIVES:

The numbers of clear objectives of input design are,

- To produce a cost effective method of input
- To achieve the highest possible level of accuracy
- To ensure that the input is acceptable to and understand by the user staff

3.3 OUTPUT DESIGN:

Output, as you probably know, generally refers to the results and information that are generated by the system. For many end-users, output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Most end users will not actually operate the information system or enter data through workstations, but they will use the output from the system.

When designing output, systems analysts must accomplish the following.

- Determine what information to present
- Decide whether to display, print, or “speak” the information and select the output medium.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to intended recipients.

That alignment of information on a display or printed document is termed as layout.

Accomplishing the general activities listed above will require specific decisions, such as whether to use preprinted forms when preparing reports and documents, how many lines to plan on a printed page, or whether to use graphics and color.

The output design is specified on layout performs, sheets that describe the location characteristics, and format of the column headings and pagination. As we indicated at the beginning of this discussion, these elements are analogous to an architect's blue print that shows the location of the each component.

3.4 DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of inter-related data stored with minimum redundancy to serve single users quickly and efficiently. The general objective is to make information necessary, quick, inexpensive and flexible for the user.

Checkout Table

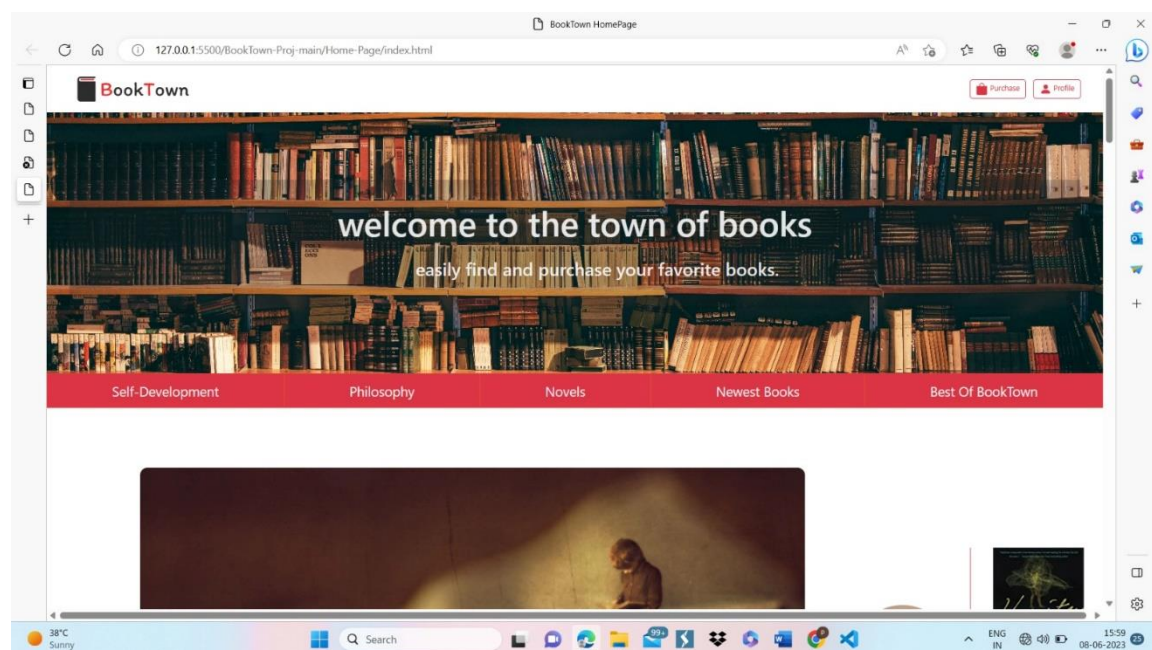
5. IMPLEMENTATION AND TESTING

5.1 Implementation approaches

The Software Design Description Document has been used as input in the implementation process. The actual implementation has been done using PHP. PHP has been used to interact with the backend database. In this implementation, My SQL Server has been used as the backend RDBMS. PHP processes the inputs or commands given by the user and translates them in the commands understandable to the backend database. The output produced by the backend database is also handled by PHP which then displayed on the Browser screen.

Coding and Screenshots

Home Page



Login Page

SignUp Page

127.0.0.1:5500/BookTown-Proj-main/SignUpModal/index.html

First name: Varsha

Last name: priya

Username: @

City: SELECT

Address: MANPUR GAYA

☐ Agree to terms and conditions

[Submit form](#) [Reset](#)

GBP/INR +0.56%




Q Search

ENG IN 19:01 08-06-2023

My Cart pages

BookTown Cart Page

127.0.0.1:5500/BookTown-Proj-main/Cart/index.html

Shopping Bag		Price	
	I Love You to the Moon and Back	+	\$4.31
	Brown Bear, Brown Bear, What Do You See?	+	\$4.55
	Verity	+	\$11.26

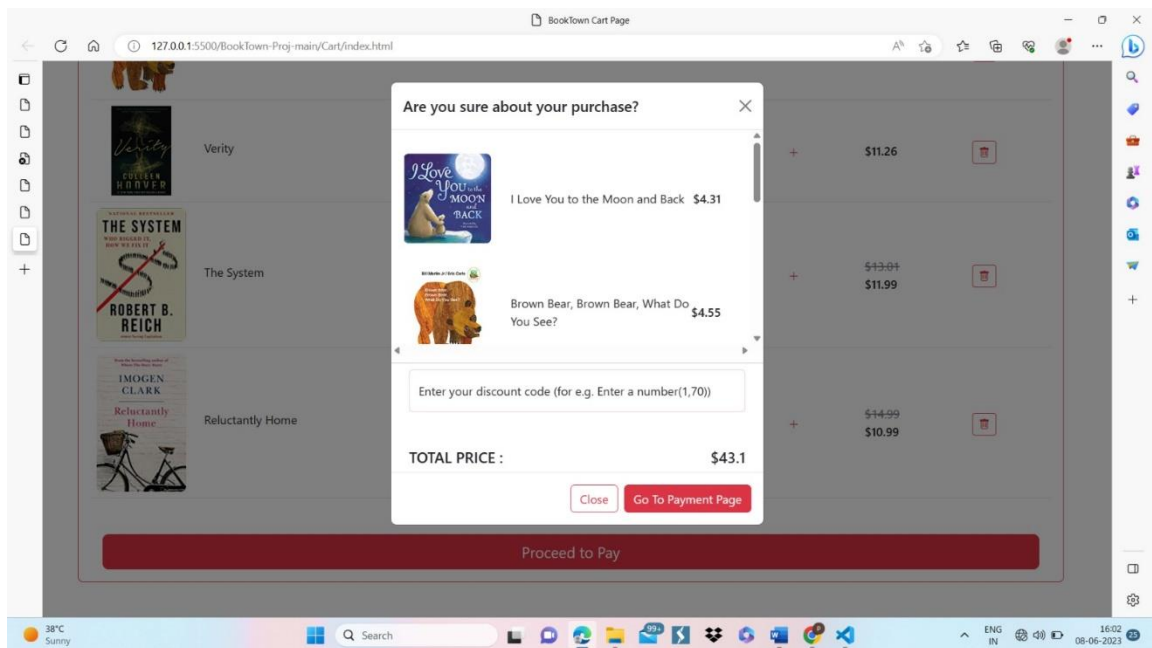
[Proceed to Pay](#)

38°C Sunny

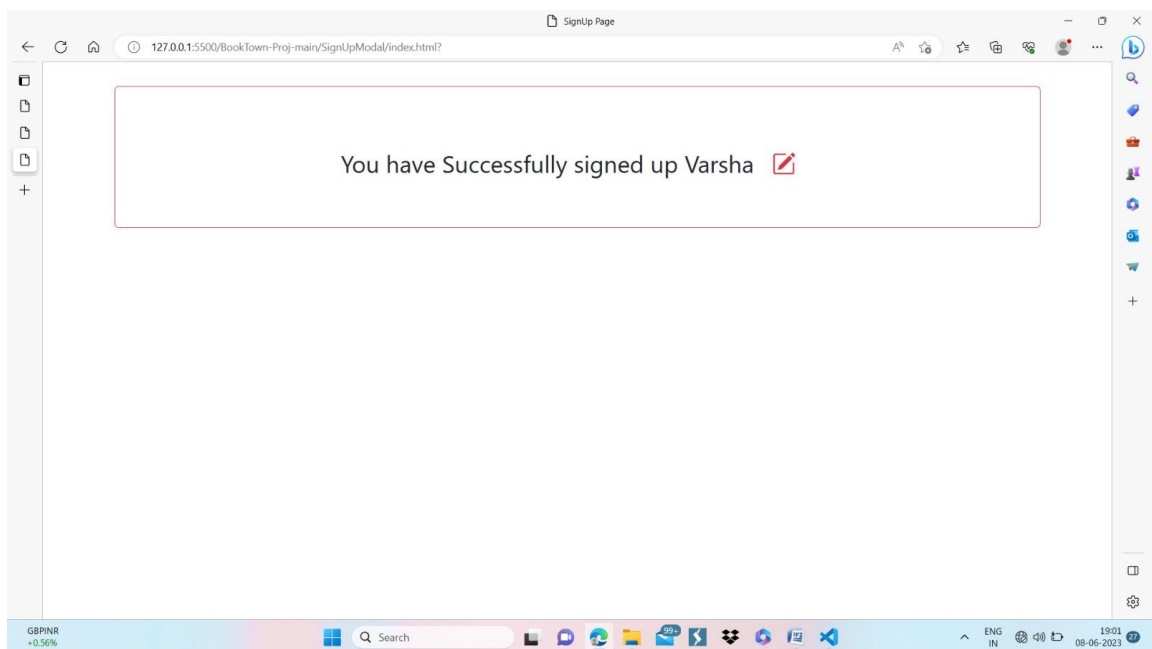
Q Search

ENG IN 16:03 08-06-2023

Payment page



Payment confirmation



Coding of index.php Page

```
<?php session_start();?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <?php
        include("includes/head.inc.php");
    ?>
</head>
<body>
    <!-- start header -->
    <div id="SearchBar">

        <?php

include("includes/searchTop.inc.php");
        ?>
    </div>
    <div id="belowHeader"> </div>
    <div>
        <div id="logo">
            <?php
                include("includes/logo.inc.php");
            ?>
        </div>
    </div>
    <div id="belowHeader"> </div>
    <div id="header">
        <div id="menu">
            <?php

include("includes/menu.inc.php");
            ?>
        </div>
    </div>
    <!-- end header -->
    <!-- start page -->
    <div id="page">
        <!-- start content -->
        <div id="content">
            <div class="post"> <!-- class post
-->
                <h1 class="title">Welcome to
Book Store

                </h1>
                <div class="entry">
                    <br>
                    <table width="95%"
border="0" align="center" cellpadding="0" cellspacing="0" id="tabCat">
                        <tr>
                            <td align="center"></td>
                            <td align="center"></td>
                        </tr>
                    </table>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

```

                                <td align="center"></td>
                                </tr>
                                <tr>
                                <td align="center">Literature &
Fiction </td>
                                <td
align="center">Architecture</td>
                                <td align="center">Art &
Culture </td>
                                </tr>
                                <tr>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                </tr>
                                <tr>
                                <td align="center"></td>
                                <td align="center"></td>
                                <td align="center"></td>
                                </tr>
                                <tr>
                                <td align="center">Sports</td>
                                <td align="center">Computers</td>
                                <td align="center">Management</td>
                                </tr>
                                <tr>
                                <td align="center">&nbsp;</td>
                                <td align="center">&nbsp;</td>
                                <td align="center">&nbsp;</td>
                                </tr>
                                <tr>
                                <td align="center"></td>
                                <td align="center"></td>
                                <td align="center"></td>
                                </tr>
                                <tr>
                                <td align="center">Entrance Exam
</td>
                                <td align="center">Law Books</td>
                                <td align="center">Comics</td>
                                </tr>
                                </table>
                                <p>&nbsp;</p>
                                </div>
                                </div>
                                </div>
<!-- end content -->

<!-- start sidebar -->
<div id="sidebar">
    <?php

```

```

include("includes/sidebar.inc.php");
                                ?>
                                </div>
                                <!-- end sidebar -->
                                <div style="clear: both;">&nbsp;</div>
                                </div>
<!-- end page -->

<!-- start footer -->
    <div id="footer">
        <?php

include("includes/footer.inc.php");
                                ?>
                                </div>
<!-- end footer -->
</body>
</html>

```

Process Login.php

```

<?php session_start();

if(!empty($_POST))
{
    $msg="";

    if(empty($_POST['usernm']))
    {
        $msg[]="No such User";
    }

    if(empty($_POST['pwd']))
    {
        $msg[]="Password Incorrect.....";
    }

    if(!empty($msg))
    {
        echo '<b>Error:-</b><br>';

        foreach($msg as $k)
        {
            echo '<li>'.$k;
        }
    }
    else
    {

        $link=mysql_connect("localhost","root","")or
die("Can't Connect...");

        mysql_select_db("shop",$link) or die("Can't Connect
to Database...");

        $unm=$_POST['usernm'];
    }
}

```



```

$q="select * from user where u_unm='$unm'";

$res=mysql_query($q,$link) or die("wrong query");

$row=mysql_fetch_assoc($res);

if(!empty($row))
{
    if($_POST['pwd']==$row['u_pwd'])
    {
        $_SESSION=array();
        $_SESSION['unm']=$row['u_unm'];
        $_SESSION['uid']=$row['u_pwd'];
        $_SESSION['status']=true;

        if($_SESSION['unm']!="admin")
        {
            header("location:index.php");
        }
        else
        {
            header("location:admin/index.php");
        }
    }

    else
    {
        echo 'Incorrect Password....';
    }
}
else
{
    echo 'Invalid User';
}
}

else
{
    header("location:index.php");
}

?>

```

Process Register.php

```

<?php

if(!empty($_POST))
{
    $msg="";

    if(empty($_POST['fnm']) || empty($_POST['unm']) ||
empty($_POST['gender']) || empty($_POST['pwd']) ||
empty($_POST['cpwd']) || empty($_POST['mail'])||empty($_POST['city']))
    {
        $msg.="<li>Please full fill all requirement";
    }
}

```

```

        if($_POST['pwd']!= $_POST['cpwd'])
        {
            $msg.="<li>Please Enter your Password Again.....";
        }

        if(!ereg("^[a-z0-9_]+[a-z0-9_.]*@[a-z0-9_-]+[a-z0-9_.-
]*\.[a-z]{2,5}$", $_POST['mail']))
        {
            $msg.="<li>Please Enter Your Valid Email Address...";
        }

        if(strlen($_POST['pwd'])>10)
        {
            $msg.="<li>Please Enter Your Password in limited
Format.....";
        }

        if(is_numeric($_POST['fnm']))
        {
            $msg.="<li>Name must be in String Format...";
        }

        if($msg!="")
        {
            header("location:register.php?error=".$msg);
        }
        else
        {
            $fnm=$_POST['fnm'];
            $unm=$_POST['unm'];
            $pwd=$_POST['pwd'];
            $gender=$_POST['gender'];
            $email=$_POST['mail'];
            $contact=$_POST['contact'];
            $city=$_POST['city'];

            $link=mysql_connect("localhost","root","")or
die("Can't Connect...");

            mysql_select_db("shop",$link) or die("Can't Connect
to Database...");

            $query="insert into
user(u_fnm,u_unm,u_pwd,u_gender,u_email,u_contact,u_city)

            values('$fnm','$unm','$pwd','$gender','$email','$contact','$city'
)";

            mysql_query($query,$link) or die("Can't Execute
Query...");

            header("location:register.php?ok=1");
        }
    }
    else
    {
        header("location:index.php");
    }
}
?>

```

Process Add_Book.php

```
<?php

if(!empty($_POST))
{
    $msg=array();
    if(empty($_POST['name']) || empty($_POST['description']) ||
empty($_POST['publisher']) || empty($_POST['edition']) ||
empty($_POST['isbn']) || empty($_POST['pages']) ||
empty($_POST['price']))
    {
        $msg[]="Please full fill all requirement";
    }
    if(!(is_numeric($_POST['price'])))
    {
        $msg[]="Price must be in Numeric Format...";
    }
    if(!(is_numeric($_POST['pages'])))
    {
        $msg[]="Page must be in Numeric Format...";
    }

    if(empty($_FILES['img']['name']))
    $msg[] = "Please provide a file";

    if($_FILES['img']['error']>0)
    $msg[] = "Error uploading file";

    if(!(strtoupper(substr($_FILES['img']['name'],-4))==".JPG"
|| strtoupper(substr($_FILES['img']['name'],-5))==".JPEG"||
strtoupper(substr($_FILES['img']['name'],-4))==".GIF"))
        $msg[] = "wrong file type";

    if(file_exists("../upload_image/".$_FILES['img']['name']))
        $msg[] = "File already uploaded. Please do not
updated with same name";

    if(empty($_FILES['ebook']['name']))
    $msg[] = "Please provide a document file";

    if($_FILES['ebook']['error']>0)
    $msg[] = "Error uploading document file";

    if(!(strtoupper(substr($_FILES['ebook']['name'],-
4))==".PDF" || strtoupper(substr($_FILES['ebook']['name'],-4))==".PPT"
||strtoupper(substr($_FILES['ebook']['name'],-5))==".PPTX" ||
strtoupper(substr($_FILES['ebook']['name'],-4))==".DOC"||
strtoupper(substr($_FILES['ebook']['name'],-4))==".TXT"||
strtoupper(substr($_FILES['ebook']['name'],-5))==".DOCX"))
        $msg[] = "wrong document file type";

    if(file_exists("../upload_ebook/".$_FILES['ebook']['name']))
        $msg[] = "Document File already uploaded. Please do
not updated with same name";

    if(!empty($msg))
    {
```

```

        echo '<b>Error:-</b><br>';

        foreach($msg as $k)
        {
            echo '<li>'.$k;
        }
    }
    else
    {

        move_uploaded_file($_FILES['img']['tmp_name'], "../upload_image/".$_FILES['img']['name']);
        $b_img = "upload_image/".$_FILES['img']['name'];

        move_uploaded_file($_FILES['ebook']['tmp_name'], "../upload_ebook/".$_FILES['ebook']['name']);
        $b_pdf = "upload_ebook/".$_FILES['ebook']['name'];

        $b_nm=$_POST['name'];
        $b_cat=$_POST['cat'];
        $b_desc=$_POST['description'];
        $b_edition=$_POST['edition'];
        $b_publisher=$_POST['publisher'];
        $b_isbn=$_POST['isbn'];
        $b_pages=$_POST['pages'];
        $b_price=$_POST['price'];

        $link=mysql_connect("localhost","root","")or
die("Can't Connect...");

        mysql_select_db("shop",$link) or die("Can't Connect
to Database...");

        $query="insert into
book(b_nm,b_subcat,b_desc,b_edition,b_publisher,b_isbn,b_page,b_price,b
_img,b_pdf)

        values('$b_nm','$b_cat','$b_desc','$b_edition','$b_publisher','$b
_isbn',$b_pages,$b_price,'$b_img','$b_pdf')";

        mysql_query($query,$link) or die($query."Can't
Connect to Query...");
        header("location:addbook.php");

    }
}
else
{
    header("location:index.php");
}
?>

```

Process Cart.php

```

<?php session_start();

        if(isset($_GET['nm']) and isset($_GET['noc']) and
isset($_GET['rate']))
        {

            function newcart()

```

```

        {
            $bnm=$_GET['nm'];
            $noc=$_GET['noc'];
            $price=$_GET['rate'];
            $uname=$_SESSION['unm'];
            $amt=$noc*$price;

            $link=mysql_connect("localhost","root","")or
die("Can't Connect...");
            mysql_select_db("shop",$link) or die("Can't Connect
to Database...");
            $cartid=rand(100,9999);
            $sql="select * from cart where cartid=$cartid";
            $result=mysql_query($sql,$link);
            $n=mysql_num_rows($result);
            if($n>1)
            {
                newcart();
            }
            else
            {
                $_SESSION['cart']=$cartid;
                $query="insert into
cart(cartid,userid,bname,price,qty,amount) values
('$cartid','$uname','$bnm','$price','$noc','$amt)";

                mysql_query($query,$link);

                //echo "Record Saved";
                header("location:viewcart.php");
            }
        }

function updatecart()
{
    $bnm=$_GET['nm'];
    $noc=$_GET['noc'];
    $price=$_GET['rate'];
    $uname=$_SESSION['unm'];
    $amt=$noc*$price;
    $cartid=$_SESSION['cart'];
    $link=mysql_connect("localhost","root","")or
die("Can't Connect...");
    mysql_select_db("shop",$link) or die("Can't Connect
to Database...");
    $query="insert into
cart(cartid,userid,bname,price,qty,amount) values
('$cartid','$uname','$bnm','$price','$noc','$amt)";

    mysql_query($query,$link);
    header("location:viewcart.php");
}

if(!isset($_SESSION['cart']) || $_SESSION['cart']=="")
{
    //create new cart id and insert the book details

    //echo "New Cart Running";
    newcart();
}

```

```

        else
        {
            //insert book details with cart id from session
            updatecart();
        }
    }
    else if(isset($_GET['id']))
    {
        //del a item
        $id = $_GET['id'];
        //unset($_SESSION['cart'][$id]);
        $link=mysql_connect("localhost","root","") or die("Can't
Connect...");
        mysql_select_db("shop",$link) or die("Can't Connect to
Database...");
        $sql="delete from cart where bno=$id";
        mysql_query($sql,$link);
        header("location: viewcart.php");
    }
}

?>

```

Process order.php

```

<?php session_start();

    if(!empty($_POST))
    {

        $link=mysql_connect("localhost","root","") or die("Can't
Connect...");
        mysql_select_db("shop",$link) or die("Can't Connect to
Database...");
        $odate=date("m/d/y");
        $cartid=$_SESSION['cart'];
        $total=$_SESSION['total'];
        $name=$_POST['fname'];
        $address=$_POST['address'];
        $discount="0";
        $status="In Progress";

        $sql="insert into
orderDetails(oDate,cardid,total,discount,fname,address,status) values
('$odate','$cartid','$total','$discount','$name','$address','$status') "
;
        mysql_query($sql,$link) or die("Can Not
Insert".mysql_error());

        echo "Thank you for your order ";
        echo '<a href="index.php">Go Back to Home Page</a>';
    }
}

?>

```

5.2 Code efficiency

Code efficiency has been achieved through proper validation using various methods in PHP coding. First no data can be added, viewed, edited and deleted to database without login or session. For this we have implanted session tracking techniques through PHP. Codings have been used to validate various forms to ensure correct data should enter in database.

5.3 Testing Approach

TESTING PROCEDURES

- **Unit Testing:** A Unit corresponds to a form/class in the package. Unit testing focuses on verification of the corresponding form or class. In this level we have tested all our forms/classes individually. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. From this testing we were able to save, retrieve, update, delete and the search records on a table.
- **Integration Testing:** Integration testing is used to verify the combination of the software modules. In this level, we have tested by combining all unit tested forms into a subsystem. Here we found that the subsystems are performing well.
- **System Testing:** System testing is used to verify, whether the developed system meets the requirements.
- **Acceptance Testing:** Acceptance is the part of the project by which the customer accepts the product. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

We hope that after the acceptance testing the system will perform the best result for the organization. When modification will be made, we will use regression testing during the maintenance of the system.

The Software System delivered to the customer may undergo changes. Changes may be due to addition of new functional modules or performance enhancement. For this purpose proper maintenance of the system is must.

6. RESULTS AND DISCUSSION

6.1 Test Reports

Test Case No	Date	Pass / Fail
TCT-001	11/5/2014	Pass
TCT-002	02/6/2014	Pass
TCT-003	25/6/2014	Pass

7. DOCUMENTATION

❖ For Management

Online Shopping System is primarily designed for providing information from the data after processing them. This system is designed for supplying information to the strategic level of management from the operational control. It includes almost all the functional areas needed like keeping Employee Records Student Records and Fees Records.

❖ For User

With this electronic data processing system, the operators will able to maintain the following task:

- Information regarding Patients.
- Records of Test Report with their details.
- Regular Transaction Details

❖ For data processing department

- In maintenance, the data processing department needs to create backup of the database file from time to time.
- The main menu of the system provides different menus for different purposes.

FUTURE APPLICATION

Software development is never –ending process and continues the life of the software as per the changing needs of the user from time to time. The project is no doubt has been developed keeping in mind easy modification and enhancement that may be required from time to time.

However, there are many scopes to modify this software. As because due to shortage of time, we here become unable to include many things. We are trying to cover all their existing system for sales return records of the items but due to shortage of time we become unable to include many things. Due to lake of time I here include none of them and a future scope one can develop these returns which are so much essential. Only with a little more doing it is possible to design the formats for those returns. Moreover, an on-line system will be more helpful to the organization. . With almost the same data with only a little modification an on-line system can be designed to fulfill their demands. All these can be considered to be future scope for this project.

CONCLUSION

After implementing the application it will contain the advantages were incomparable to the present contemporary systems used by company. The most admirable feature founded was its simplicity in terms of application to the user but its highly beneficial outputs can't be ignored. The users will be highly benefited after using the system.

It is hoped that this project will help the future developers to modify and implement the system. After modifying some techniques of the programs, it will give us the best performance as our requirements. The project will be very useful for the users.