COOK-D-BOOK

**A**

**MINOR PROJECT-II REPORT**

Submitted in partial fulfillment of the requirements

for the degree of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

By

**GROUP NO. = 51**

##### SAGAR GUPTA-0187CS181121

**AKSHAY TURKAR-0187CS181020**

**WAJAHAT KHAN-0187CS181170**

**RASHAD AHEMAD-0537CS181019**

Under the guidance of

**Umesh Kumar Sahu**

(Assistant Professor)

****

**June-2021**

**Department of COMPUTER SCIENCE & ENGINEERING**

**Sagar Institute of Science & Technology (SISTec)**

**Bhopal (M.P.)**

**Approved by AICTE, New Delhi & Govt. of M.P.**

**Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**

***Sagar Institute of Science & Technology (SISTec), Bhopal***

## *Department of* *COMPUTER SCIENCE & ENGINEERING*

***Bhopal (M.P.)***

******

***June-2021***

# *CERTIFICATE*

I hereby certify that the work which is being presented in the B.Tech. Minor Project-II Report entitled **COOK-D-BOOK,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science & Engineering*** and submitted to the Department of Computer Science & Engineering, *Sagar Institute of Science & Technology (SISTec)****,*** Bhopal (M.P.) is an authentic record of my own work carried out during the period from Jan-2021 to June-2021 under the supervision of **Proff. Umesh Kumar Sahu.** The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

|  |  |  |
| --- | --- | --- |
| **Akshay Turkar** | **0187CS181020** |  |
| **Sagar Gupta** | **0187CS181170** |  |
| **Wajhat Khan** | **0187CS181170** |  |
| **Rashad Ahmed** | **0537CS181019** |  |

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

***Date:***

**TABLE OF CONTENTS**

***Principal***

***HOD***

***Project Guide***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TITLE** | | | | **PAGE NO.** |
|  | | | |  |
| Abstract | | | | 4 |
| Acknowledgement | | | | 5 |
| List of tables | | | | 6 |
| List of figures | | | | 7 |
| List of abbreviations | | | | 8 |
| Chapter 1 | Introduction | | | 9 |
|  | 1.1 | | Outline of project |  |
|  | 1.2 | | Project Overview |  |
| Chapter 2 | Software & Hardware Requirements | | | 13 |
| Chapter 3 | Problem Description | | | 18 |
| Chapter 4 | Literature Survey | | | 19 |
| Chapter 5 | Software Requirements Specification | | | 20 |
|  | 5.1 | | Functional Requirements |  |
|  | 5.2 | | Non-Functional Requirements |  |
| Chapter 6 | Software Design | | | 23 |
|  | 6.1 | | ER Diagram |  |
|  | 6.2 | | Table Structure |  |
| Chapter 7 | Output Screens | | | 37 |
| Chapter 8 | Future applications | | | 41 |
| References | | | | 43 |
| Project Summary | | | | 48 |
|  | |
|  | |
|  | |

**ABSTRACT:**

Cook-D-Book is an Online book Management System is a system which maintains the information about the second hand books present on the website, their authors, the members of library to whom books are bought and sell. This is very difficult to organize manually. Maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of selling and buying second hand books becomes much simple. The Cook-D-Book has been designed to computerize and automate the operations performed over the information about the members, book selling and buying and all other operations. This computerization of this management helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. A special gratitude I give to my respected guide Prof. UMESH KUMAR SAHU department in computer science in SISTec, whose contribution in stimulating suggestions and encouragement helped me to coordinate my project especially in writing this report.

Furthermore I would also like to acknowledge and thanks to the principle Dr. Keshvendra Chouhdary, Bhavna Gupta, HOD of the computer science department, all faculty members and staff for providing me all the facilities and for their support to all activities, and gave the permission to use all required equipment and the necessary materials to complete the Project.

I would like to express my gratitude to my parents for their unbreakable believe, support and guidance. Last but not the least I would like to thanks all my classmates of B.Tech Computer science batch 2018-2022 for their co-operation and support.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE OF TABLE** | **PAGE NO.** |
| 2.1 | LITERATURE SURVEY | 17 |
| 6.4.1 | USER TABLE | 36 |
| 6.4.2 | BUY BOOKS TABLE | 36 |
| 6.4.3 | SELL BOOKS TABLE | 37 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG. NO.** | **TITLE** | **PAGE NO.** |
| 6.2.3 | CONTEXT DIAGRAM | 25 |
| 6.2.4 | DFD FOR ADMIN PROCESS | 26 |
| 6.2.5 | FOR USER REGISTRATION AND PROFILE UPDATE | 26 |
| 6.2.6 | DFD FOR SHOPPING AND CHECKOUT PROCESS | 27 |
| 6.2.7 | DFD FOR SELLING AND CHECKOUT PROCESS | 28 |
| 6.3.2 | ER DIAGRAM | 32 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ACRONYM** | **FULL FORM** |
| CSS | Cascading Style Sheet |
| MY-SQL (lite) | MY - Structured Query Language |
| HTML5 | Hypertext Markup Language 5 |
| JS | Java Script |
| PHP | Hypertext Preprocessor |
| BOOTSTRAP | Bootstrapv4.6 |
| VS-CODE | Visual Studio Code |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**CHAPTER 1**

* 1. **INTRODUCTION**

The project titled COOK-D-BOOK is book selling software for monitoring and controlling the Ads of buying and selling books. The project “COOK-D-BOOK” is developed in php, which mainly focuses on basic operations in a library like selling and buying second hand books, and updating new information, searching books and members and return books.

This project of “COOK-D-BOOK” of gives us the complete information about the library. We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records and can also check how many books are issued and stock available in the library.

Throughout the project the focus has been on presenting information and comments in an easy and intelligible manner. The project is very useful for those who want to get books.

* + 1. **OUTLINE OF PROJECT REPORT**

The motivation to create this project has many sources

•Interest to develop a good user friendly website with many online transactions using a database.

•To increase my knowledge horizon in technologies like PHP, SQL, CSS, HTML5, BOOTSTRAP and JAVASCRIPT.

•To gain good experience in PHP before joining in a full time job.

•To gain expertise using Data Grid, Data Set, Data Table, Data Adapter and Data Readers

**1.2 PROJECT OVERVIEW**

There are many COOK-D-BOOKs like Powell’s, Amazon which were designed using Html. I want to develop a similar website using PHP, SQL Server. COOK-D-BOOK is an online web application where the customer can purchase and sell books online.

Through a web browser the customers can search for a book by its title or author and finally purchase using credit card transaction. The user can login using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The books are divided into many categories based on subject Like Software, Database, English, Architecture etc. And based on genre like Fiction, Mystery, Action, Love etc.

This project has the following functionalities:

1. **A Home page with product catalog**

This is the page where the user will be navigated after a successful login. It will display all the book categories and will have a search keyword option to search for the required book. It also includes some special sections like recommended titles, weekly special books.

1. **Search**

A search by keyword option is provided to the user using a textbox .The keyword to be entered should be the book title.

**3) Advanced Search**

Advanced search helps the user to search for a book based on Title, Author, Category and price range. All the books which match the particular search criteria and their total count will be displayed .From here the user can select a book and add to the shopping cart.

**4) Book Description**

If the user would like to know details about a book he can click on the title from where he will be directed to a Book description page. It includes the notes on the book content and also a link to Amazon.com to get the book review.

5**) Managing user accounts**

Each user should have an account to access all the functionalities of website. User can login using login page and logout using the logout page. All the user sessions will be saved in the database.

**6) User Access**

The User will be provided with special functionalities like

•Add or delete a book advertisement

•Manage book orders.

•Add or delete a Credit Card type or other payment options.

**1.3 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 User will regularly add any new books available to them for sale. The User will sell second hand and also buy second hand books.

**1.4 PURPOSE AND SCOPE**

**1.4.1 PURPOSE**

**COOK-D-BOOK** would have the following goals.

* Provide a web user interface to add, view, delete records in different areas.
* 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 php technologies. This knowledge will be useful in the future in creating any type of desktop application or online database systems.

**CHAPTER 2**

**2.1 SOFTWARE & HARDWARE REQUIREMENTS**

In a desktop application like COOK-D-BOOK, 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
* Bootstrap

**Server Side Scripting:** PHP

**Database Tool:** MySQLi

**Testing Server:** Apache

**Other Software Used:**

* Netbeans
* VS- Code

**HTML**

**HTML** or **Hyper Text 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 <img>. 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/).

**CHAPTER 3**

**3.1 PROBLEM DESCRIPTION**

**Problem Description and Need for the New System**

* COOK-D-BOOK 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 “**COOK-D-BOOK”.**

**CHAPTER 4**

**4.1 LITERATURE SURVEY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S No.** | **Paper title** | **Author** | **Year** | **Output** | **Limitations** |
| 1 | Demography, Motivation, and Satisfaction of E-Commerce Customer on OLX.co.id and Tokopedia.com | Henry Hendrayati1 Vanessa Gaffar2 , and Marcela Atrisia3 | 2015 | Can sell all your second hand components | They don’t have scope for academic books |
| 2 | Developing second hand Electronics | Syed EmdadUllah, Tania Alauddin and Hasan U. Zaman | 2016 | Can Buy and Sell second hand electronic components | There is lack of Good UI and Not option for books. |
| 3 | Book Chor app on playstore | Vidyut Sharma and team | 2016 | We can buy and sell books online | We have to sell books to store of book chor and buy from them |

**CHAPTER 5**

**3.1 REQUIREMENTS AND SPECIFICATION**

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 behaviors.
* 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.

**3.1.2 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.
* Users are using for viewing and buying the books.
* User(seller) can add, edit & delete books and provide services to the customer.
* User(seller) can see the daily sell. Can also see the feedback given by the customer.
* Users maintaining the deliveries.

**3.1.3 FUNCTIONAL REQUIREMENTS**

* The System must provide following functionalities—
* Keeping records of registration of customers.
* Keeping the records of books.
* Maintaining the chats and calls between the users.
* Adding books with description, price and genre.
* Shorting the books by their genre.

**3.1.4 NON FUNCTIONAL REQUIREMENTS**

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

* Secure access of confidential data (user’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

**3.1.5 EXTERNAL INETRFACE 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:-

Frontend- HTML5, CSS3, BOOTSTRAP, JAVASCRIPT

Backend- PHP, Ajax

Database- MySQL, PhpMyadmin

Server- Apache

**CHAPTER 6**

* 1. **SOFTWARE DESIGN**

**6.1.1 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

**6.1.2 MODULES DESCRIPTION**

1. Registration: Users can register their account here to continue buying/selling.
2. Users: Users can add books, buy books, check chats, contact with buyer.
3. Payment: Cash on Delivery facility is available.
4. Users can contact the buyer(user) for the delivery of the book they can choose place to meet .

**6.1.3 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.

* **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

**6.1.4 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.

**6.2 CONCEPTUAL MODELS**

**6.2.1 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).

* blank**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 that 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 show that the systems do. Each process has one or more data inputs and one or 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 sequent 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 retrieved data from the data store. An open rectangle is a data store, data at rest.

**-Level DFD:**

Order receipt & confirmation

Order receipt & confirmation for

Personal Details & Select product or sell product for user 2

Personal Details & Select product or sell product for user

User 2

User 1

6.2.3 **CONTEXT DIAGRAM**

Data

User

Status

Book

Product

Product

Store

Access/Upload

process

Buy

**6.2.4** DFD FOR USERS PROCESS

Enter data

View Profile

Edit data

User Details

Register

User

Status

User

Updates

**6.2.5 DFD For User Registration and Profile Update**

Order Details

Removes add

Confirms

Chat Details

Remove ad

Selected Product

Book info

Login

User

Select Product or post an ad

Checkout

Sold

Book

Chats

User

**6.2.6** DFD FOR SHOPPING AND CHECKOUT PROCESS

Book added

Order Details

Confirmation & receipt

Confirms

Cart Details

Selected Product

Book info

Login

User

Post an ad

Ad posted

Book added

Book

My Ads

User

**6.2.7** DFD FOR SELLING AND CHECKOUT PROCESS

# 6.3.2 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

Entitiesare 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 entityis one that does not rely on another for identification. A dependent entityis 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

Attributesdescribe 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 nary 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 relationshipis one in which one of the child entities is also a dependent entity. A non-identifying relationshipis one in which both entities are independent.

## Existence

Existencedenotes 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

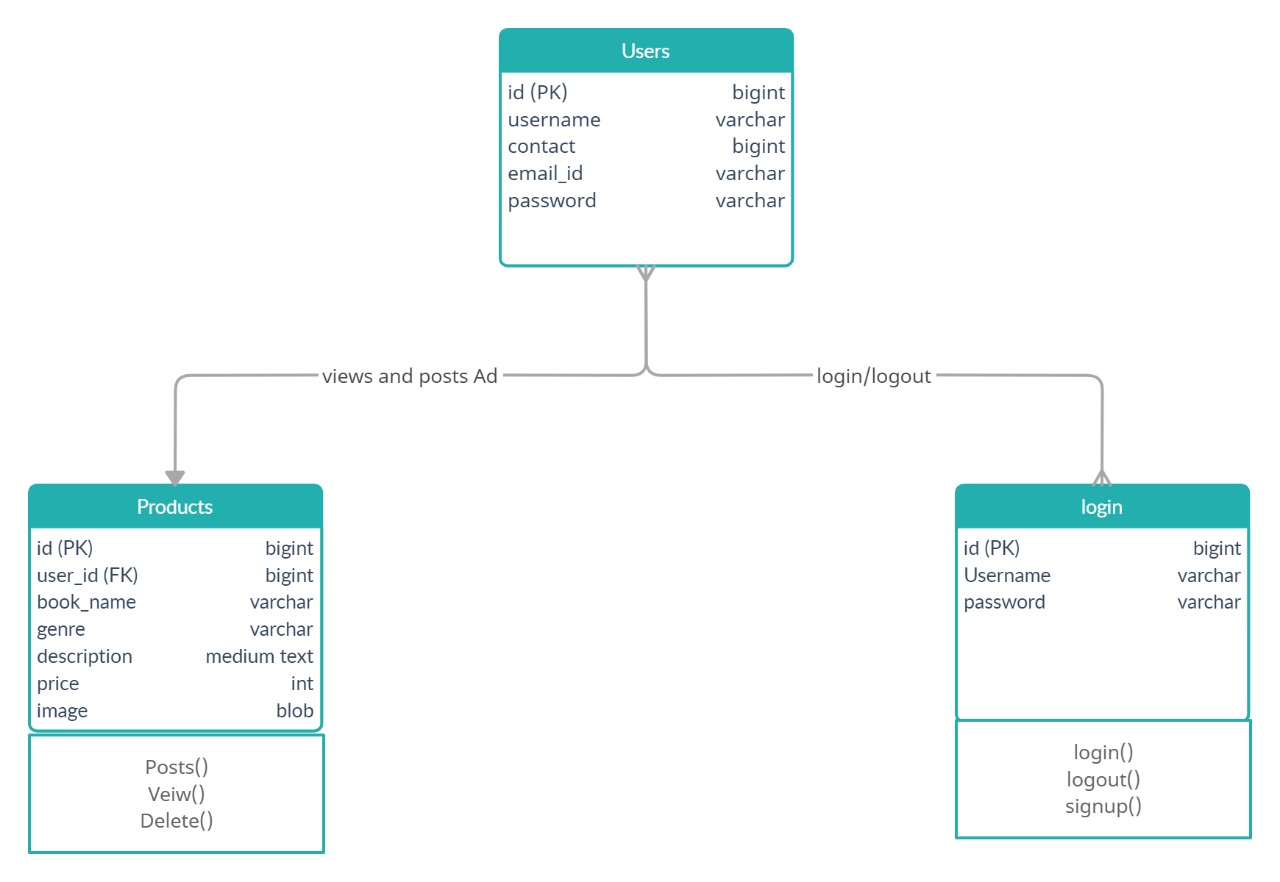
A generalization hierarchy is a form of abstraction that specifies that two or more entities that share common attributes can be generalized into a higher level entity type called a super typeor genericentity. The lower-level of entities become the subtype, or categories, to the super type. Subtypes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

****

**6.3.2 ER diagram6.3.3 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.

**6.4 Database Tables**

**6.4.1 User Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Allow Null** | **Constrain** |
| Username | varchar | 255 | No | PK |
| Email\_id | varchar | 255 | No |  |
| Password | varchar | 255 | No |  |
| Contact | Bigint | 10 | No |  |
| id | Bigint | 11 | No |  |

**6.4.2 Products Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Allow Null** | **Constrain** |
| Id | bigint | 11 | No | PK |
| Bookname | varchar | 255 | No |  |
| Genre | varchar | 255 | No |  |
| price | float | (10,2) | No |  |
| Description | mediumtext | 500 | No |  |
| User\_id | bigint | 11 | No | FK |

**CHAPTER 7**

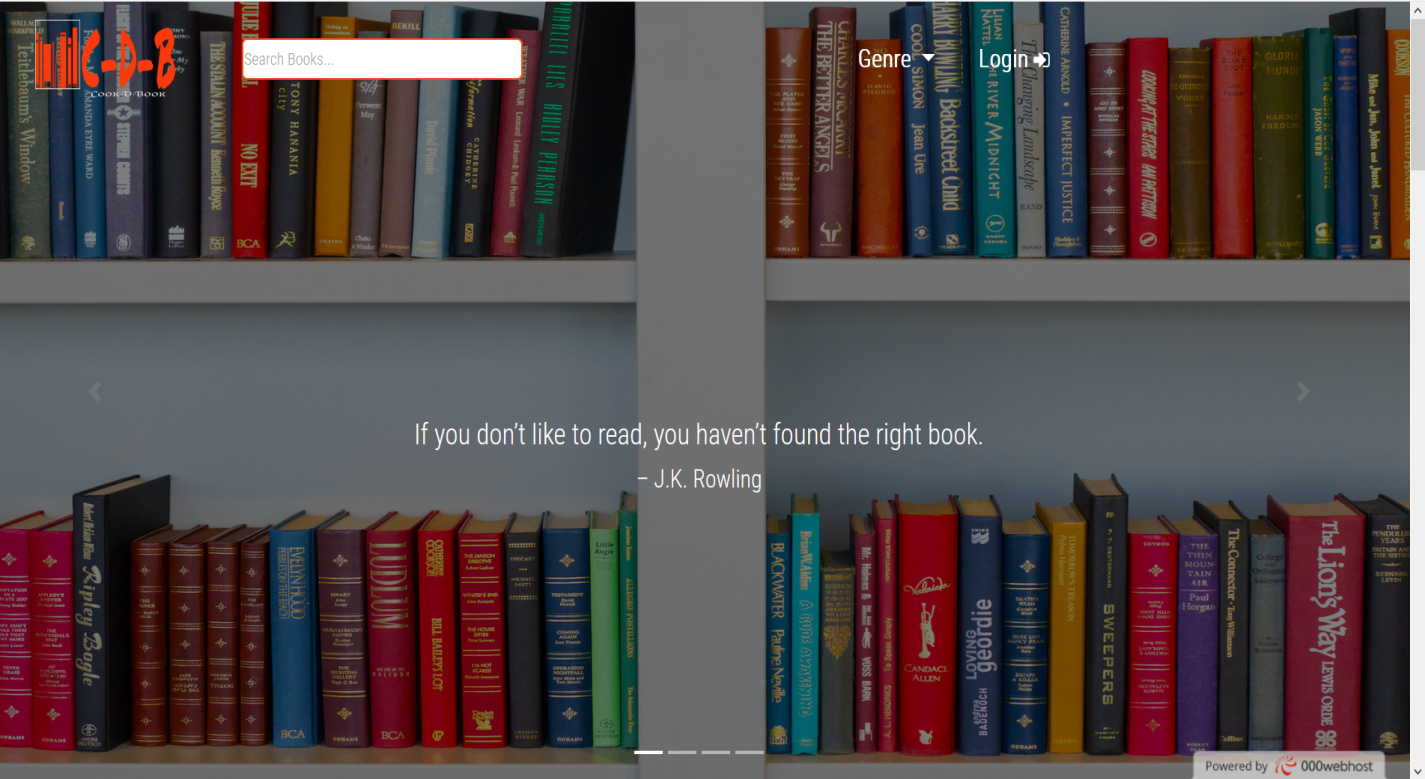
**7.1 OUTPUT**

**7.1.1 IMPLEMENTATION APPROACHES**

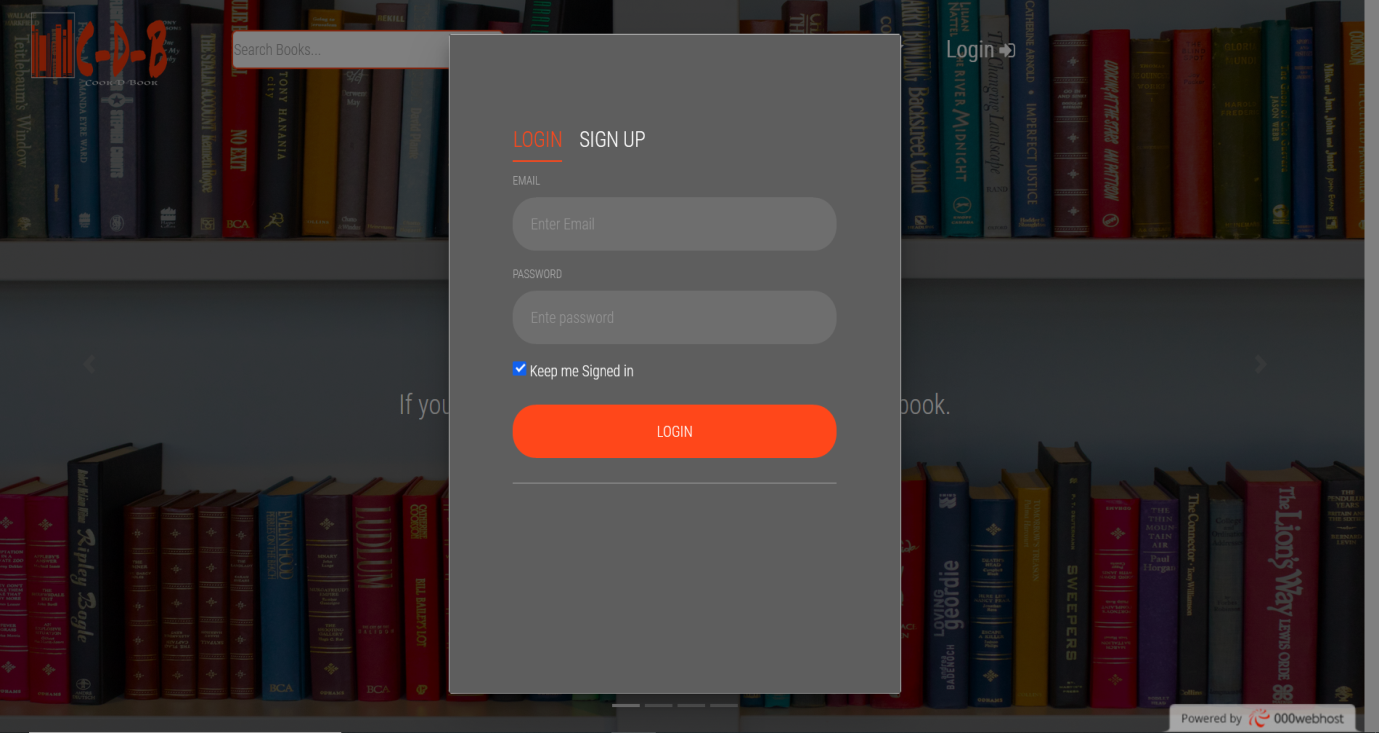
The Software Design Description Document has been used as input in the implementation process. The actual implementation has been done using PHP. PHP has been used to interact with the backend database. In this implementation, My SQL Server has been used as the backend RDBMS. PHP processes the inputs 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.

7.1.2 CODING AND SCREENSHOTS

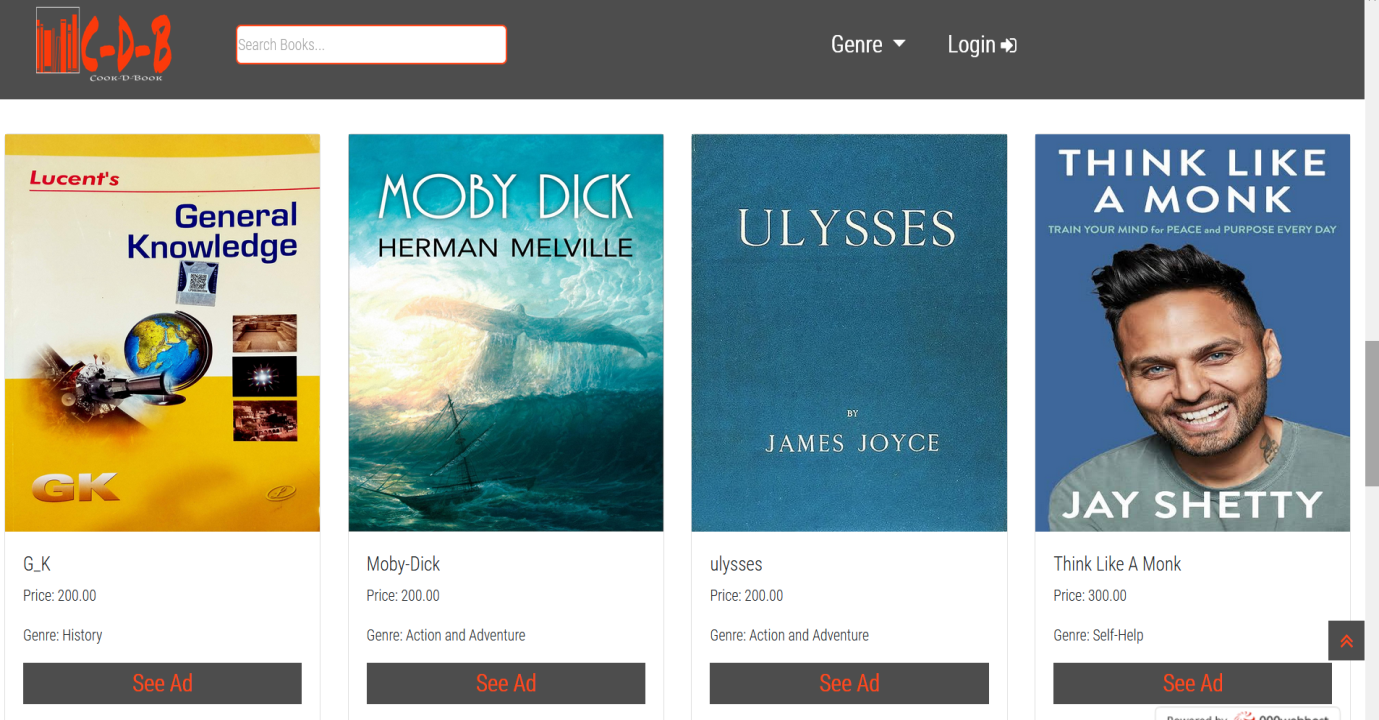
**Index page**

****

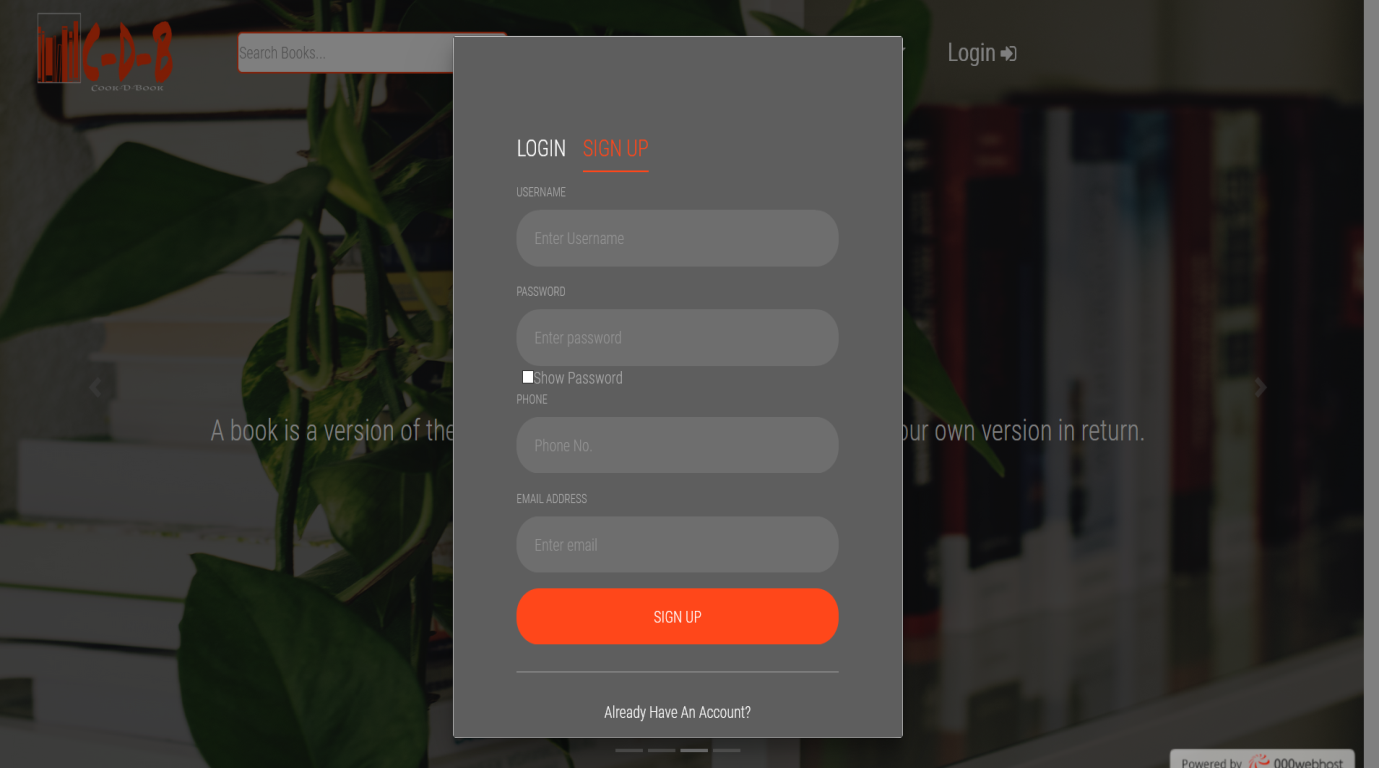
**Login page**

****

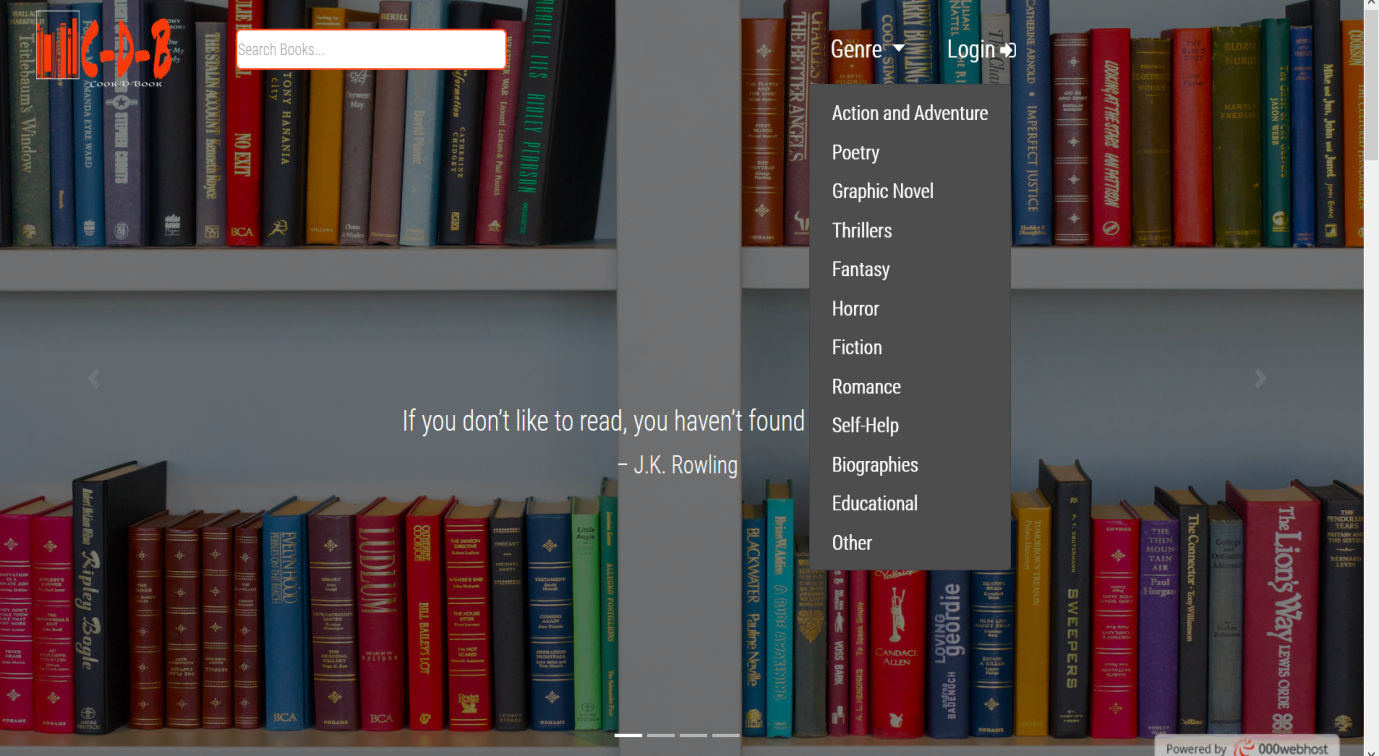
**Book List Page**

****

**Signup Page**

****

**Category Selection**

****

**7.2 CODE EFFICIENCY**

Code efficiency has been achieved through proper validation using various methods in PHP coding. Fist 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. Codlings have been used to validate various forms to ensure correct data should enter in database.

**CHAPTER 8**

**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.

**REFERENCES**

**Books:**

**1:** High Performance MySQL: Optimization, Backups, Replication, and More- Author : [Derek J. Balling](https://www.google.com/search?sxsrf=ALeKk018nYg27CMFVnMdfGheTMTtMfad3g:1622221877725&q=Derek+J.+Balling&stick=H4sIAAAAAAAAAOPgE-LRT9c3NEoqLsorTDFX4gXxDNOTs5IMLTLStGSyk630k_Lzs_XLizJLSlLz4svzi7KtEktLMvKLFrEKuKQWpWYreOkpOCXm5GTmpe9gZQQAC9KPgFMAAAA&sa=X&ved=2ahUKEwjOsZC27-zwAhU97XMBHfO1DoIQmxMoATAYegQIBBAD)

**Websites:-**

**1:** [**https://www.w3schools.com/html/default.asp**](https://www.w3schools.com/html/default.asp)

**2:** [**https://www.w3schools.com/bootstrap4/default.asp**](https://www.w3schools.com/bootstrap4/default.asp)

**3:** [**https://www.w3schools.com/php/default.asp**](https://www.w3schools.com/php/default.asp)

**4:** [**https://www.w3resource.com/sql/tutorials.php**](https://www.w3resource.com/sql/tutorials.php)

**PROJECT SUMMARY**

***About Project***

|  |  |
| --- | --- |
| **Title of the project** | COOK-D-BOOK |
| **Semester** | VI |
| **Members** | 4 |
| **Team Leader** | Akshay turkar |
| **Describe role of every member in the project** | Akshay turkar – backend, database  Sagar gupta – front end  Wajahat khan – designing  Rashad ahemad - database |
| **What is the motivation for selecting this project?** | To improve the literacy rate for poor students |
| **Project Type**  **(Desktop Application, Web Application, Mobile App, Web)** | Web application |

***Tools &Technologies***

|  |  |
| --- | --- |
| **Programming language**  **used** | PHP , CSS3, HTML5, BOOTSTRAP4.6,JAVASCRPIT, MY-SQL |
| **Compiler used** | Google console. |
| **IDE used**  **(with version)** | VS CODE(1.56.2) |
| **Front End Technologies**  **(with version, wherever**  **Applicable)** | HTML5, CSS3, JAVASCRIPT, BOOTSTRAP v4.6 |
| **Back End Technologies**  **(with version, wherever**  **applicable)** | PHP 7.02 |
| **Database used**  **(with version)** | MY-SQL (lite) |

***Software Design& Coding***

|  |  |
| --- | --- |
| **Is prototype of the software developed?** | no |
| **SDLC model followed**  **(Waterfall, Agile, Spiral etc.)** | Waterfall |
| **Why above SDLC model is followed?** | Because it simple and very easy to use and follows linear-sequential life cycle model. |
| **Justify that the SDLC model mentioned above is followed in the project.** | As Waterfall model is linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin similar as we applied in our project. First we have developed frontend the adding functionalities using backend and then we have created database and tables using MySQL. There was no overlapping in the phases. |
| **Software Design approach followed** | We have used functional Software design approach. |
| **No. of Tiers** | 3-tiers. |
| **Total no. of front end pages** | 4 |
| **Total no. of tables in database** | 2 |
| **Database is in which Normal Form?** | 2NF |
| **Are the entries in database encrypted?** | Yes the password in database is sha1 encrypted. |
| **Front end validations applied** | Yes |
| **Session management done** | Yes session set after successful signup or login. |
| **Is application browser compatible** | Yes the website is full responsive and as well as compatible. |
| **Exception handling done** | No |
| **Commenting done in code** | Yes for easy readability. |
| **Naming convention followed** | Yes CamelCase convention is followed. |
| **What difficulties faced during deployment of project?** | Hard to find free hosting server. |
| **Total no. of Use-cases** | 4 |
| **Give titles of Use-cases** | Sign up  Login  Add books  Delete books |

***Project Requirements***

|  |  |
| --- | --- |
| **MVC architecture followed** | No. |
| **Design Pattern used** | No |
| **Interface type** | GUI |
| **No. of Actors** | 2 |
| **Name of Actors** | User(buyer), User(seller) |
| **Total no. of Functional**  **Requirements** | 9 |
| **List few important non-**  **Functional Requirements** | Ajax dynamic search bar, Secure and Password encryption, Scalable, Usability. |

***Testing***

|  |  |
| --- | --- |
| **Which testing is performed?** | Manual testing. |
| **Is Beta testing done for this**  **project?** | No |

***Write project narrative covering above mentioned points***

|  |
| --- |
| The project titled COOK-D-BOOK is book selling software for monitoring and controlling the Ads of buying and selling books. This project is made using Waterfall SDLC method since it is an easy and linear-sequential life cycle model. The goal of project is to Provide a web user interface to add, view, delete records in different areas. Provide a user interface to change details of all the computers and accessories. The scope of this project is Understand and prepare detailed requirement and specifications . Prepare high level and detailed design specifications of the system |

|  |  |
| --- | --- |
| Akshay Turkar | Guide Signature |
| 0187CS181020  Sagar Gupta  0187CS181121  Wajahat Khan  0187CS181170  Rashad Ahemad  0537cs181019 | (prof. Umesh Kumar Sahu) |

**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.