# OLD BOOK STORE

**A PROJECT REPORT**
**Submitted By**

**SHIKHA CHAUDHARY**
**(University Roll No: 2000290140112)**
**SHREYA GAUR**
**(University Roll No: 2000290140115)**
**ARSHI GOEL**
**(University Roll No: 2000290140026 )**
**SIMRAN SAIFI**
**(University Roll No: 2000290140121 )**

**Submitted in partial fulfillment of the**
**Requirements for the Degree of**

# MASTER OF COMPUTER APPLICATIONS

**Under the Supervision of**
**Mr. Naresh Chandra**
**ASSISTANT**
**PROFESSOR**
**KIET Group of Institution, Ghaziabad**

**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**
**KIET Group of Institutions, Ghaziabad**
**Uttar Pradesh-201206**

**(JAN 2022)**

# CONTENTS

**TITLE PAGE**

**CANDIDATE DECLARATION** **i**

**CERTIFICATE** **ii**

**ABSTRACT** **iii**

**ACKNOWLEDGMENT** **iv**

**CHAPTER: 1 INTRODUCTION** **1-7**

1.1 Objective ..... 3

1.2 Scope ..... 3

1.3 Initial Investigation ..... 5

1.4 Technology Selection ..... 6

1.5 Overview of Advance Java ..... 7

**CHAPTER: 2 REQUIREMENTS AND ANALYSIS** **8-16**

2.1 Problem Definition…. ..... 9

2.2 Requirements Specification ..... 10

2.3 Software and Hardware Requirements ..... 13

2.2.1 Hardware Requirements……..

2.2.2 Software Requirements……..

2.4 Preliminary Product Description ..... 13

2.5 Conceptual Models ..... 1

**CHAPTER: 3 IMPLEMENTATION AND TESTING** **17-26**

3.1 Implementation Approaches ..... 1.8

3.2 Coding Details and code Efficency ..... 19

3.1.1 Coding Details

3.1.2 Code Efficiency

3.3 Entity Relationship Diagram ..... 25

3.4 Database Tables ..... 26

**CHAPTER: 4 TESTING AND RESULT ANALYSIS 27-29**

4.1 Methodology used for testing ..... 28

4.2 Testing Methods..... 28

**CHAPTER: 5 CONCLUSION AND FUTURE WORK 30-31**

5.1 Conclusion… ..... 31

5.2 Future Enhancement ..... 31

**REFERENCES**

# CANDIDATE DECLARATION

We hereby declare that the work, which is being presented in the project, entitled **Old Book Store** towards the partial fulfillment of the requirement for the award of the degree of **MASTER of COMPUTER APPLICATION** in the department of **KIET Group of Institution, Ghaziabad (India)** is an authentic record of my own work carried out during the period from October 2021 to December 2021, under the guidance of **Mr. Naresh Chandra, Assistant Professor KIET Group of Institution, Ghaziabad .I have not** submitted the matter embodied in this project for the award of any other degree or diploma.

**SHIKHA CHAUDHARY(2000290140112)**
**SHREYA GAUR (2000290140115)**
**ARSHI GOEL (2000290140026)**
**SIMRAN SAIFI (2000290140121)**

Date:15 jan 2022
Place: Ghaziabad

# CERTIFICATE

Certified that **SHIKHA CHAUDHARY(University Roll No: 2000290140112) SHREYA GAUR(University Roll No: 2000290140115) ARSHI GOEL(University Roll No: 2000290140026) SIMRAN SAIFI (University Roll No: 2000290140121)** have carried out the project work having "**OLD BOOK STORE**" for Master of Computer Applications from **Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow** under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**SHIKHA CHAUDHARY(2000290140112)**
**SHREYA GAUR (2000290140115)**
**ARSHI GOEL (2000290140026)**
**SIMRAN SAIFI (2000290140121)**

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

Date: 15 JAN 2022

**Mr. Naresh Chandra**
**Associate Professor**
**Department of Computer Applications**
**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner** **Signature of External Examiner**

**Dr. Ajay Shrivastava**
**Head, Department of Computer Applications**
**KIET Group of Institutions, Ghaziabad**

# ABSTRACT

It is an online website where customer can buy and sell second hand books .Through web browser customer can search for the book,later add to the cart anf finally purchase it.

The books are divided into categories to make the search easier.The user give feedback about the books.

The main objective is to provide a second hand book at convenient price.To save the time of the users by providing books online.

To provide service to distant areas.

User can buy used book according to their need.

# ACKNOWLEDGEMENT

I express my deep sense of gratitude to **Mr. Vipin Kumar, Associate Professor Department of Computer Applications KIET Group of Institutions, Ghaziabad**. Whose kindness valuable guidance and timely help encouraged me to complete this volume on a very crucial issue related to the work and who helped me in completing the project and he exchanged his interesting ideas, thoughts and made this project easy and accurate.

I express my thanks to **Mr. Ankit Verma** for extending his support.

I express my thanks to **Dr. Ajay Shriwastav**, **Head, Department of Computer Applications KIET Group of Institutions, Ghaziabad**.for kindly intention.

I express my thanks to the authors whose works have been consulted by me during the project.

I would also thank my institution and my faculty members without whom this project would have been a distant reality.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way without whom I would be unable to complete my project .

At last but not the least I want to thank my friends who appreciated me for my work and motivated me and finally to god who made all the things possible.

**SHIKHA CHAUDHARY (2000290140112)**
**SHREYA GAUR(2000290140115)**
**ARSHI GOEL (2000290140026)**
**SIMRAN SAIFI (2000290140121)**

# CHAPTER - 1

INTRODUCTION

# Introduction

**Old Book Store** is the process whereby consumers directly buy old books. from a seller interactively in real-time without an intermediary service over the internet.

Old book store is the process of buying books 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 OBJECTIVE

It is an online website where customer can buy and sell second hand books .Through web browser customer can search for the book,later add to the cart anf finally purchase it.

The books are divided into categories to make the search easier.The user give feedback about the books.

The main objective is to provide a second hand book at convenient price.To save the time of the users by providing books online.

To provide service to distant areas.

User can buy used book according to their need.

## 1.2 SCOPE

Purpose

**Old book Store** 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.

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.

## Description of the project:

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.

## 1.3 INITIAL INVESTIGATION

The first phase of software project is to gather requirements .Gathering software requirements begins as a creative brainstorming process in which the goal is to develop an idea for a new and modules that no other software vendor has thought. New software modules ideas normally developed as a result of analyzing the project

The main function of requirements gathering phase is to take an abstract idea that fills a particular needs or that solves a particular problems and create a real world project with a particular sets of objectives, timeline and team.

Some of the highlights of the requirements gathering phase include:

- Collecting project ideas.
- Gathering customer requirements and proposed solution.
- Justifying the project.
- Submitting the request for proposal
- Getting the team in phase.
- Preparing the requirements documents.

### Collecting project ideas:

coming up with project ideas can prove expansion exercise.

### Problem Definition

The problem is to provide the complete information about the college campus. In which the college staff members and students can access the information and will be familiar with college campus. It will provide interactive environment for the staff and students by getting knowledge of attendance, remarks, exams performances, grades, exam-timetables, notices etc.

## 1.4 TECHNOLOGY SELECTION

MERN (Mongodb,Express,React,Nodejs)

We select the Mern for our project

### Java offers the following features:

1. Compiled and Interpreted.
2. Platform-Independent and Portable.
3. Object-Oriented.
4. Robust and Secure.
5. Distributed
6. Architecture-Neutral.
7. Familiar and Simple.
8. Multithreaded and Interactive.
9. High Performance.
10. Dynamic and Extensible

## 1.5 OVERVIEW OF ADVANCE JAVA

Java is divided into two parts i.e. Core Java (J2SE) and Advanced Java (JEE). The core Java part covers the fundamentals (data types, functions, operators, loops, thread, exception handling, etc.) of the Java programming language. It is used to develop general purpose applications. Whereas Advanced Java covers the standard concepts such as database connectivity, networking, Servlet, web-services, etc.

The four major benefits of advance Java that are, network centric, process simplification, and futuristic imaging standard.

- JEE (advance Java) provides libraries to understand the concept of **Client-Server architecture** for web- based applications.
- We can also work with web and application servers such as **Apache Tomcat** and **Glassfish** Using these servers, we can understand the working of HTTP protocol. It cannot be done in core Java.
- It is also important understand the advance Java if you are dealing with trading technologies like **Hadoop, cloud-native** and **data science**.
- It provides a set of services, **API** and **protocols**, that provides the functionality which is necessary for developing **multi-tiered** application, web-based application.
- There is a number of advance Java frameworks like, **Spring, Hibernate, Struts,** that enables us to develop secure **transaction-based** web applications such as banking application, inventory management application.

## CHAPTER - 2

# CONCEPT AND TECHNIQUES

## 2. REQUIREMENTS AND ANALYSIS

## 2.1 Problem Definition

## Problem Definition and Need for the New System

- Old 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 "**Old 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

## 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:

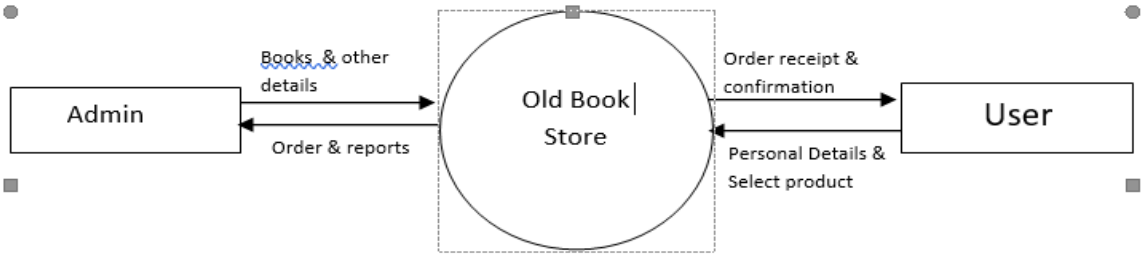
- External entities are outside to system but they either supply input data in the system or use the system output. These are represented by square of rectangle. External entities that supply data into a system are sometimes called Sources. External entities that use system data are sometimes called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

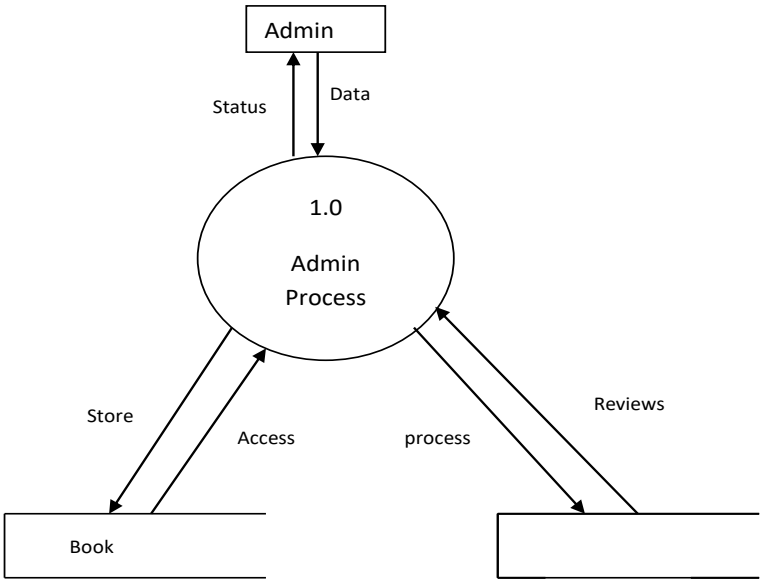
  - Receivable process.
  - Verifiable process.
  - Disposal process.

- 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.

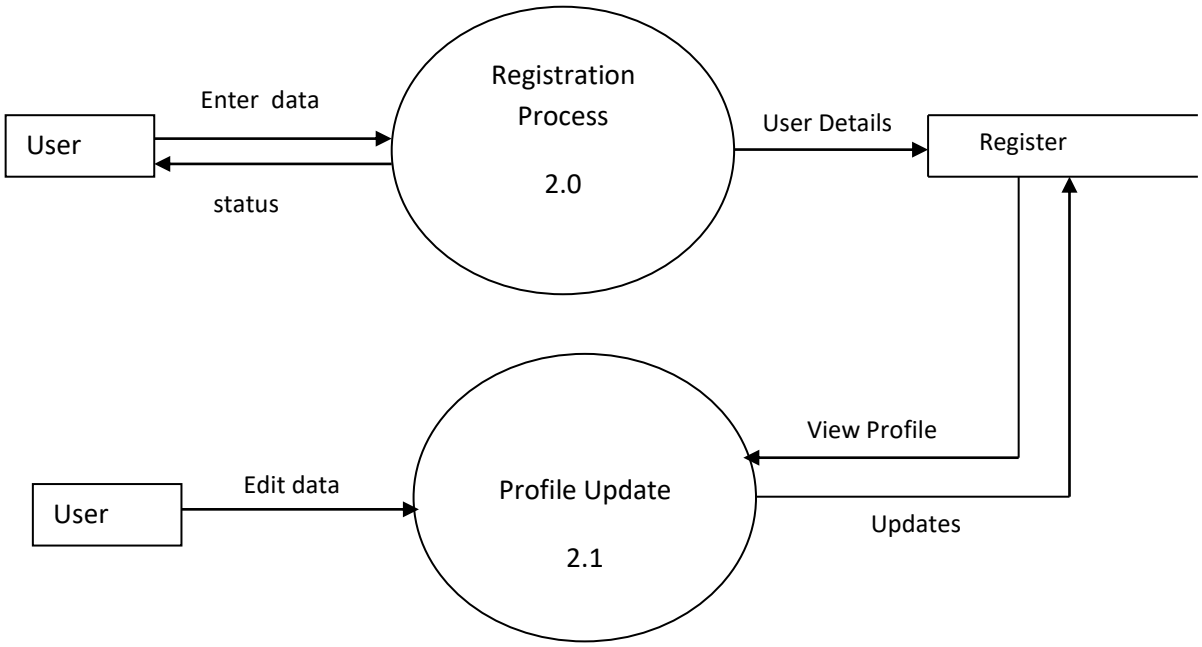
**0-Level DFD:**



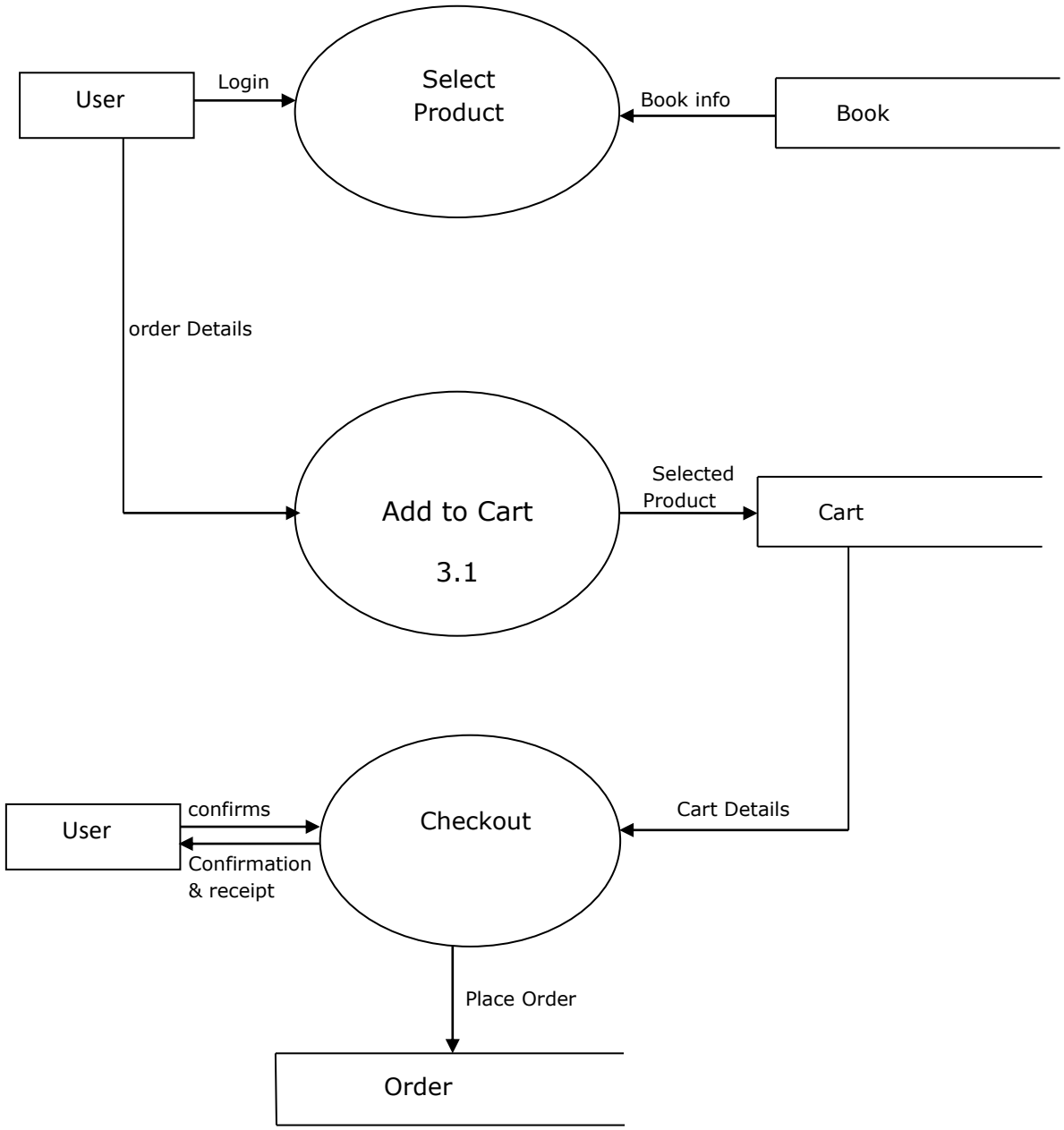

**CONTEXT DTAGRAM**




## 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.

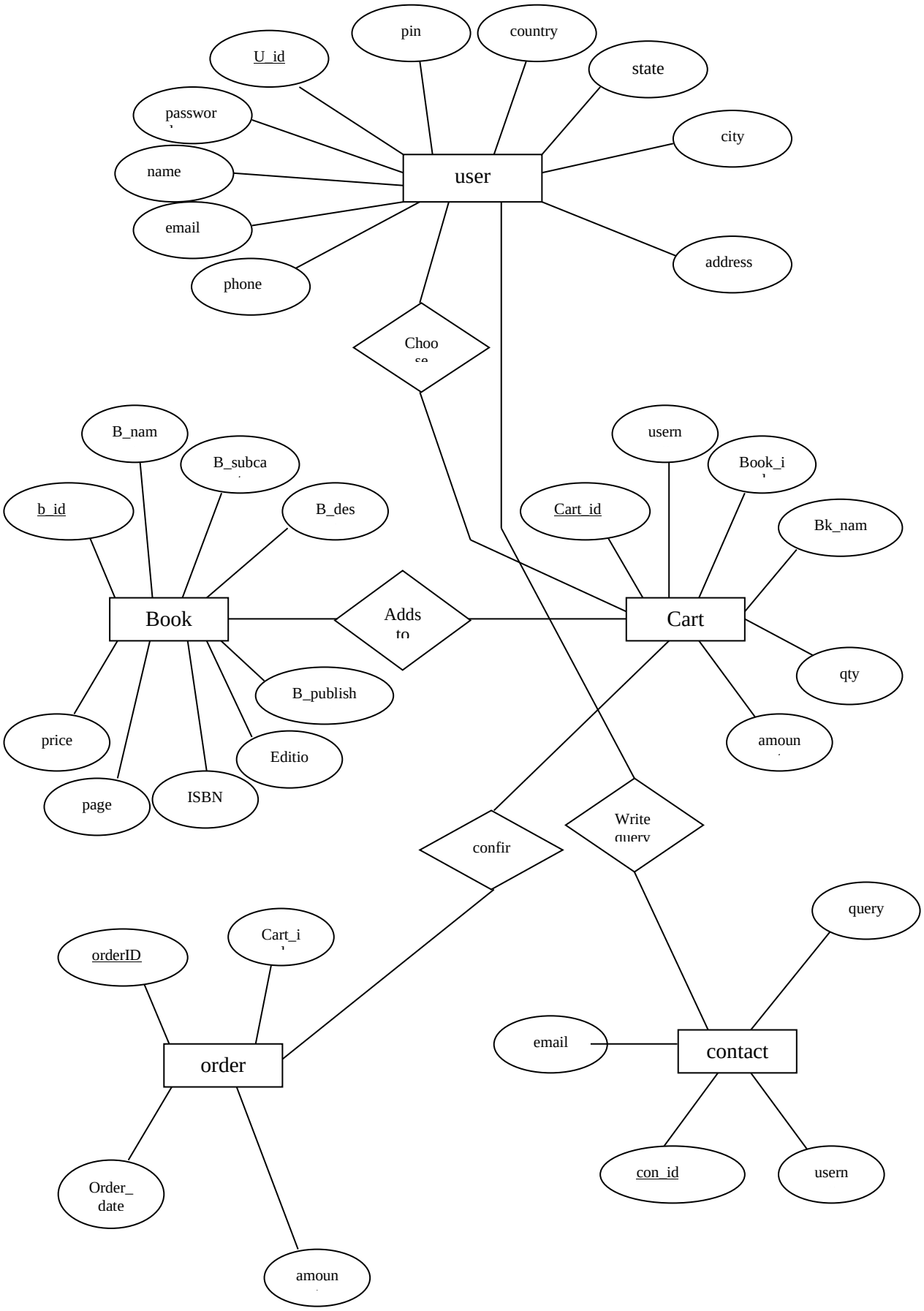
## Generalization Hierarchies

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.


pin
country
U_id
state
passwor
city
name
user
email
address
phone
Choo
B_nam
usern
B_subca
Book_i
b_id
B_des
Cart_id
Bk_nam
Book
Adds
to
Cart
qty
B_publish
price
amoun
Editio
page
ISBN
Write
query
confir
query
Cart_i
orderID
email
order
contact
Order_
date
con_id
usern
amoun

+

## Chapter - 3

# Implement And Testing

# 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.

## . REQUIREMENTS AND ANALYSIS

## 2.1 Problem Definition

## Problem Definition and Need for the New System

- Old 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 "**Old 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

## 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:

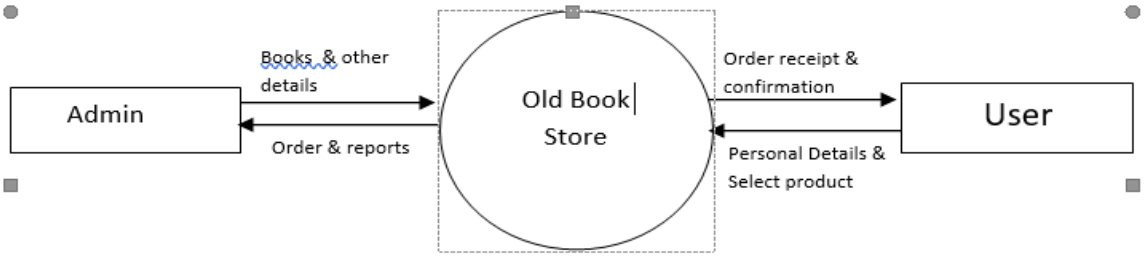
- External entities are outside to system but they either supply input data in the system or use the system output. These are represented by square of rectangle. External entities that supply data into a system are sometimes called Sources. External entities that use system data are sometimes called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

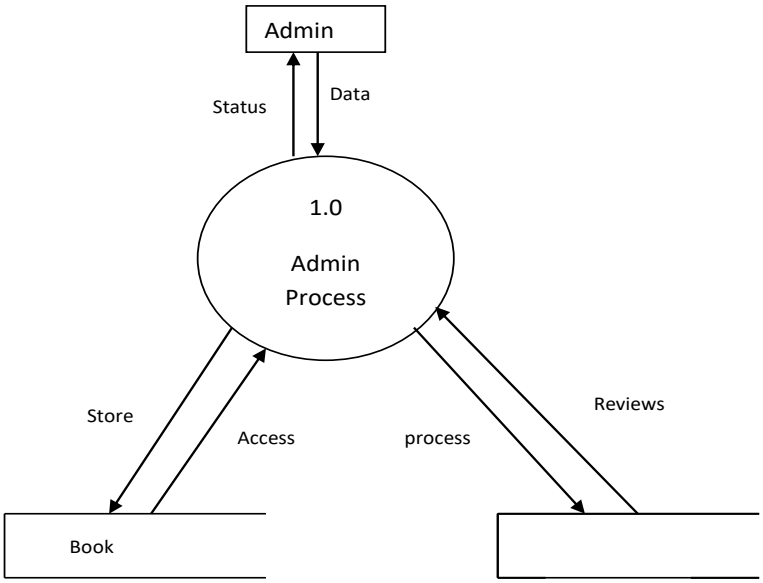
  - Receivable process.
  - Verifiable process.
  - Disposal process.

- 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.

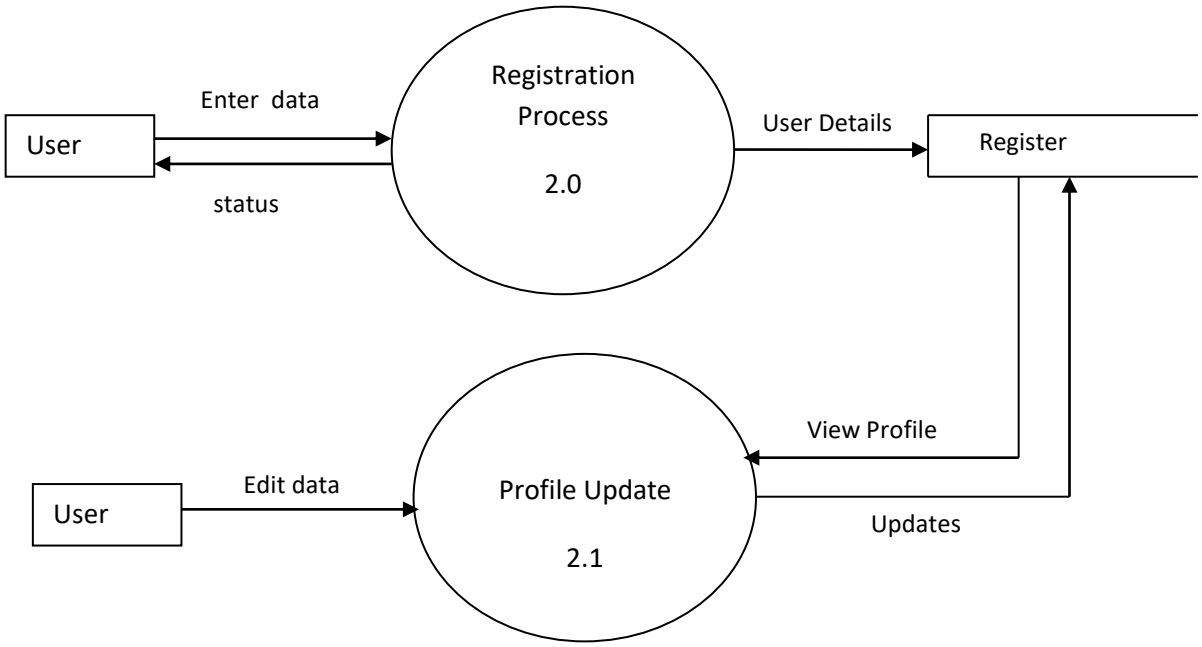
**0-Level DFD:**



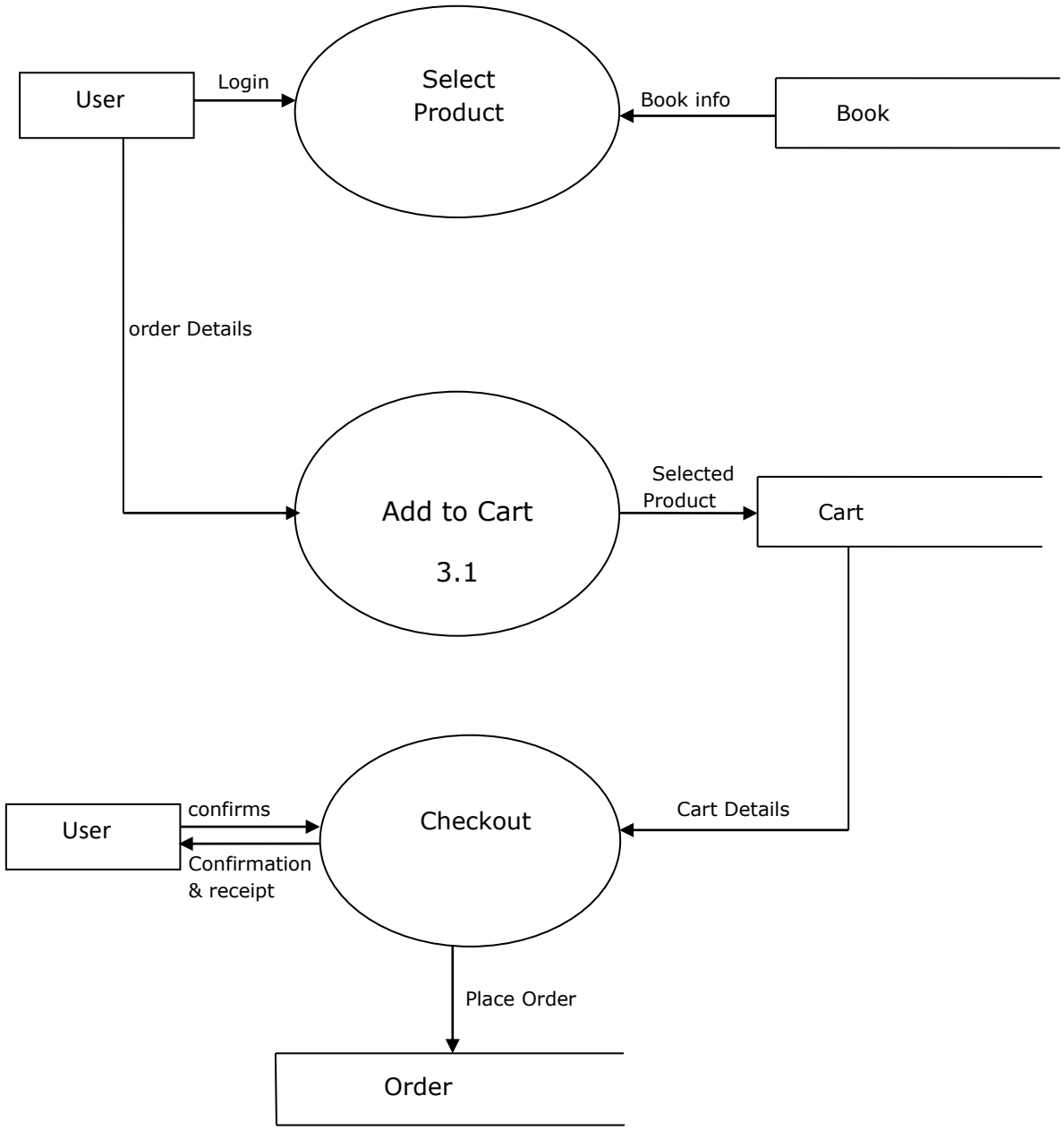

**CONTEXT DTAGRAM**




## 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.

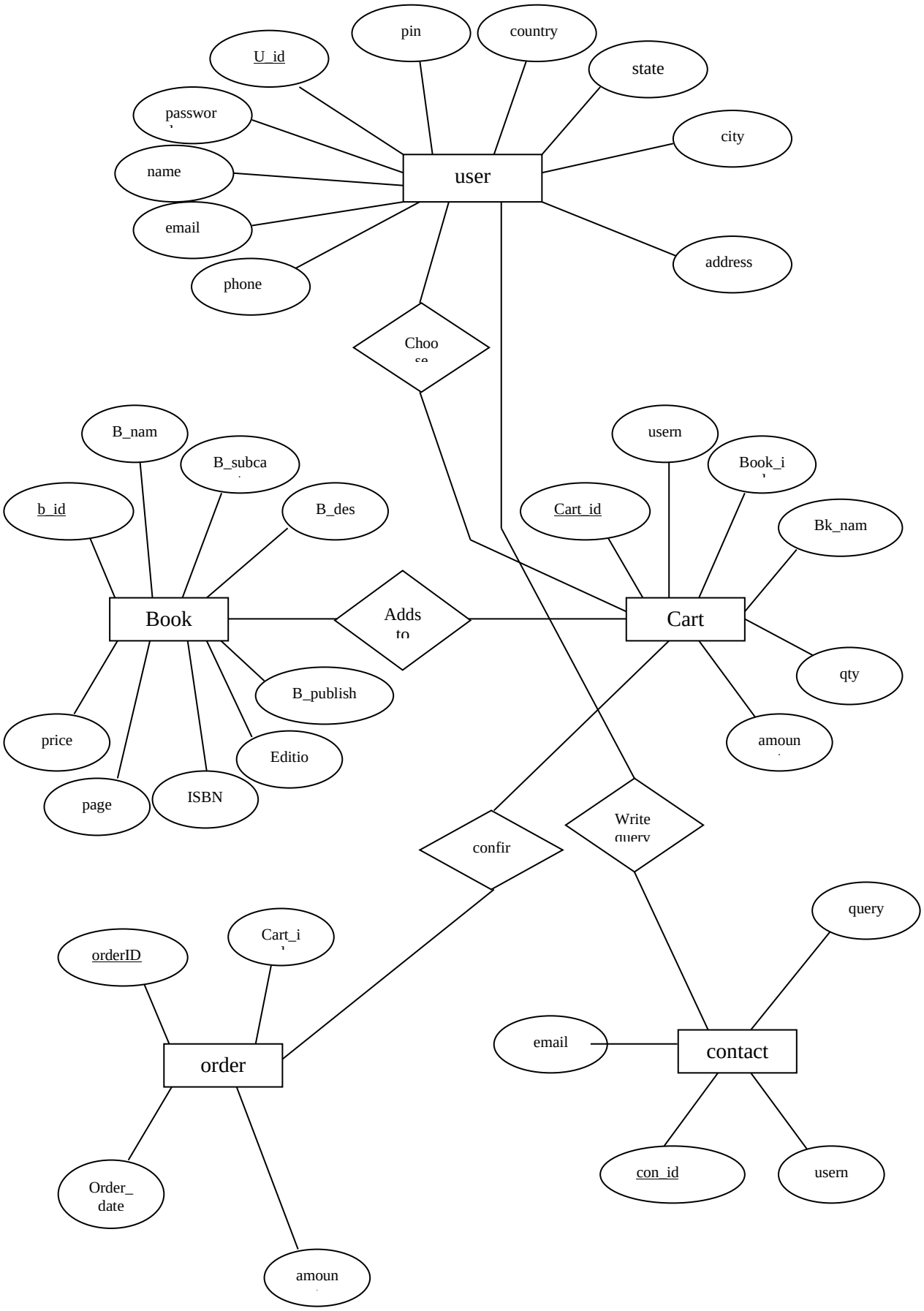
## Generalization Hierarchies

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.


pin
country
U_id
state
passwor
city
name
user
email
address
phone
Choo
B_nam
usern
B_subca
Book_i
b_id
B_des
Cart_id
Bk_nam
Book
Adds
to
Cart
qty
B_publish
price
amoun
Editio
page
ISBN
Write
query
confir
query
Cart_i
orderID
email
order
contact
Order_
date
con_id
usern
amoun

+

## Chapter - 3

# Implement And Testing

# 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.

## 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

## 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.

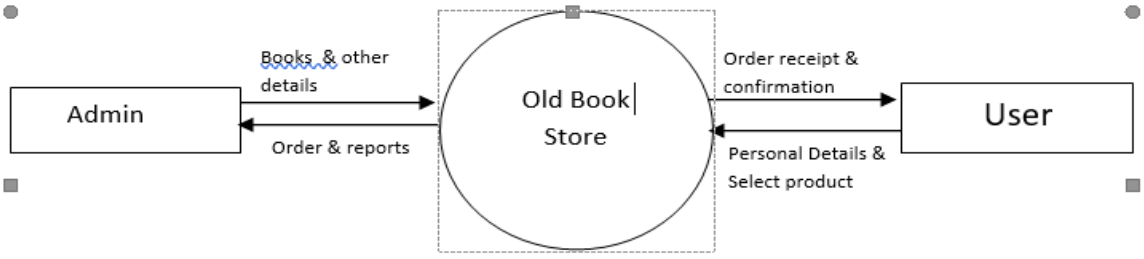
- Dataflow models that passages of data in the system and are represented by line by joining system components. An arrow indicates the direction of the flow and the line is labeled by the name of the dataflow.
- Process show that the systems do. Each process has one or more data inputs and one or data 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:

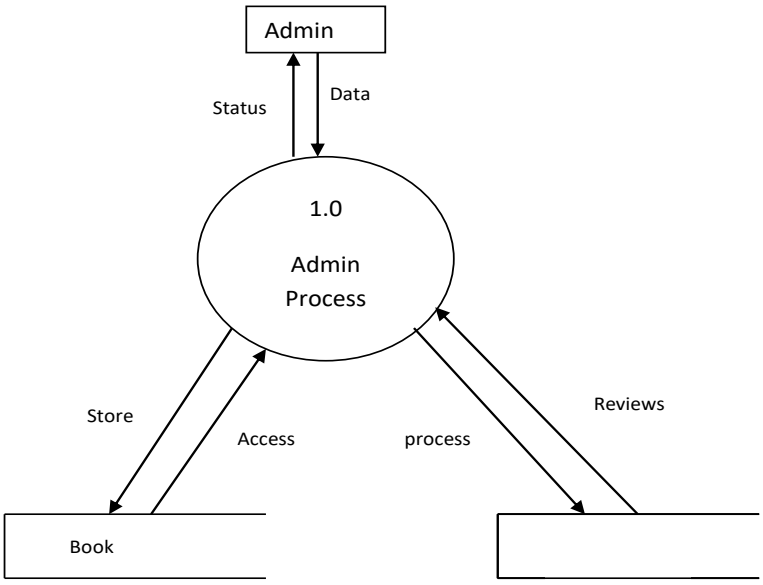
  - Receivable process.
  - Verifiable process.
  - Disposal process.

- 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.

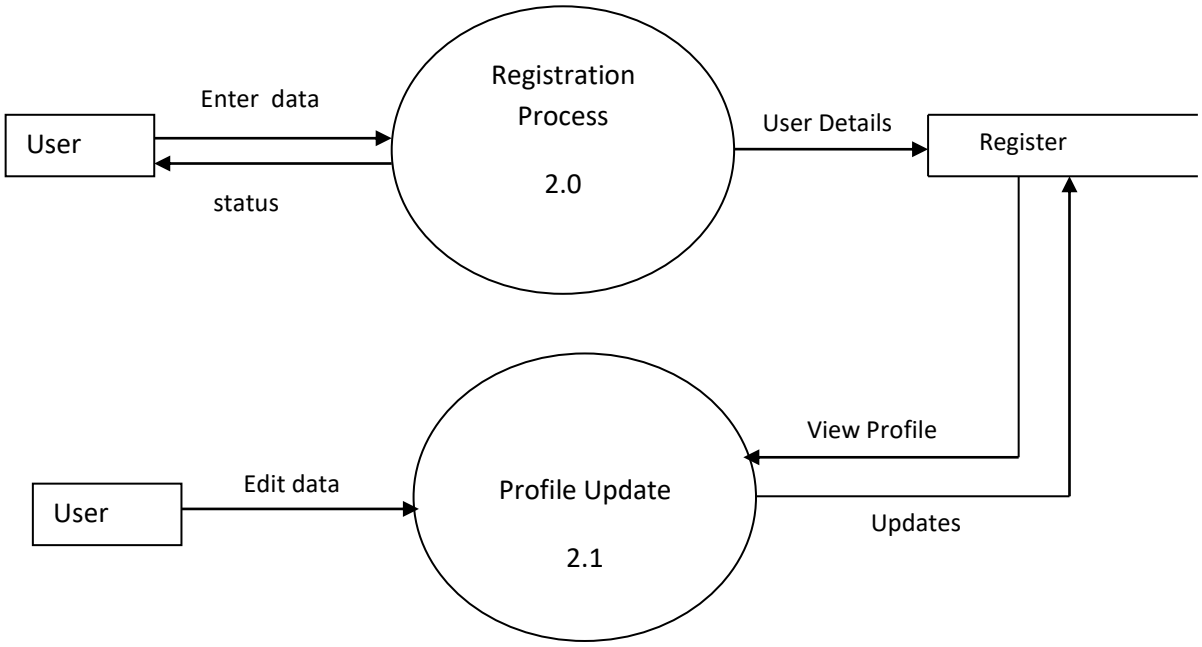
**0-Level DFD:**



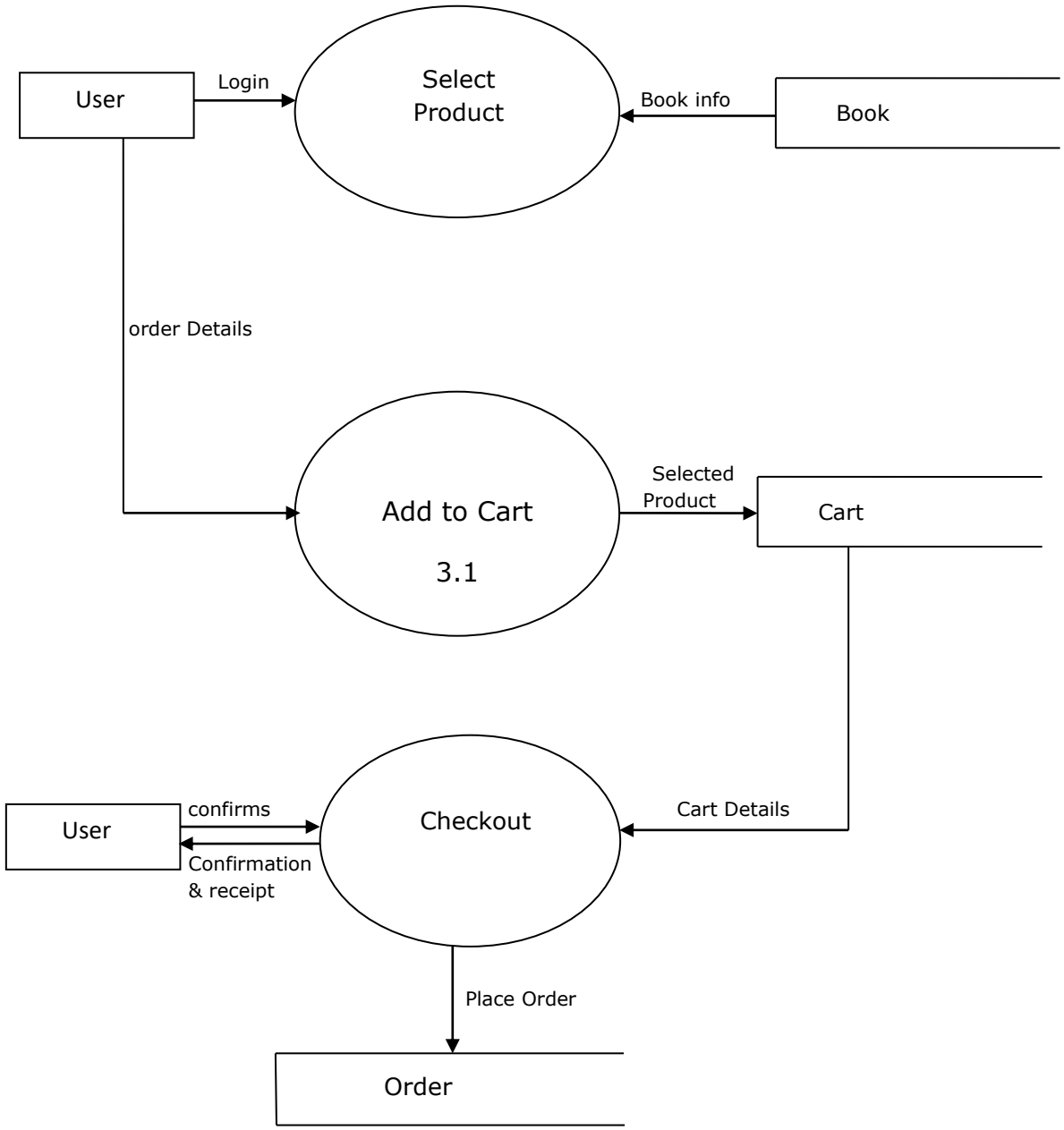

**CONTEXT DTAGRAM**




## 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.

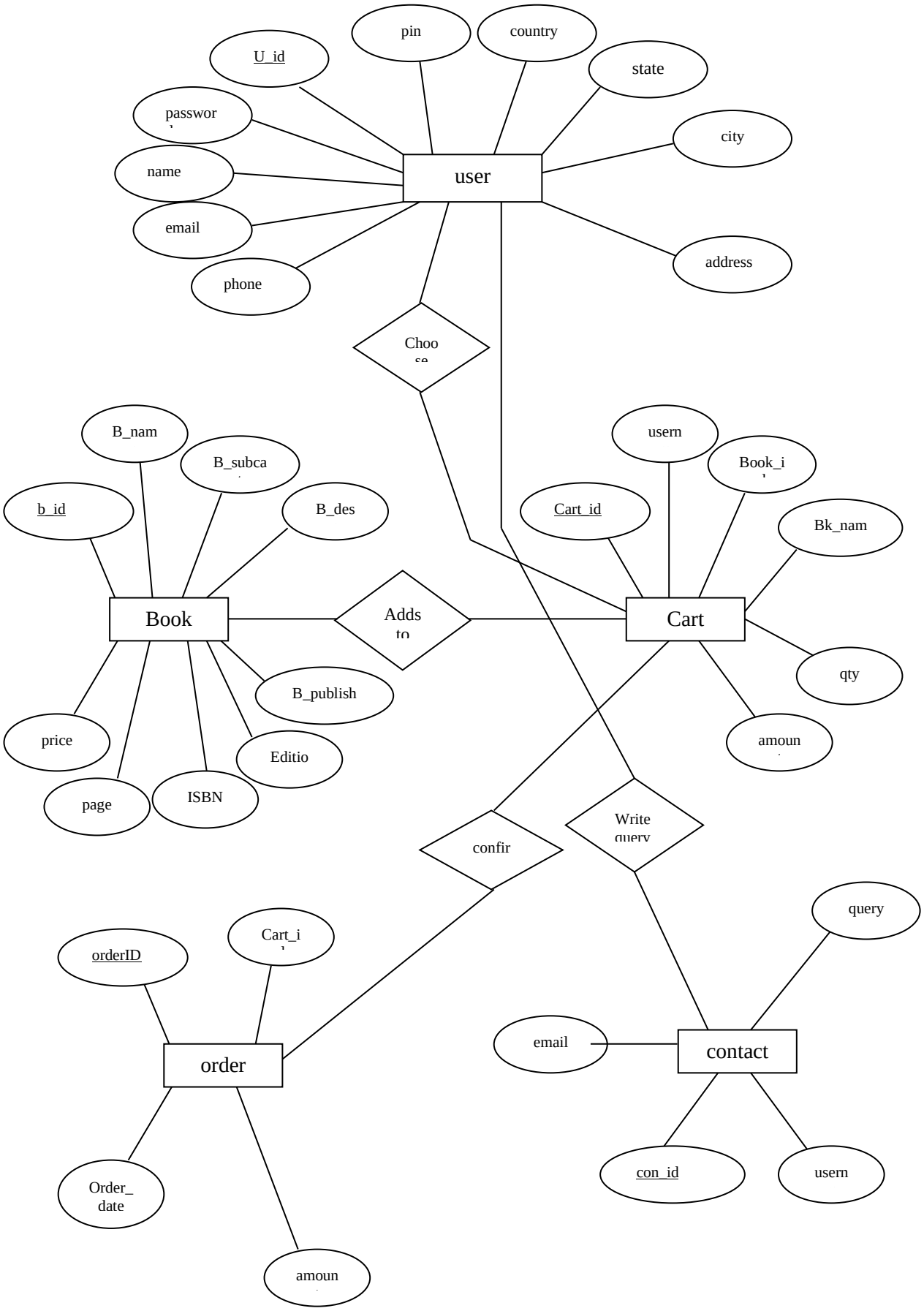
## Generalization Hierarchies

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.


pin
country
U_id
state
passwor
city
name
user
email
address
phone
Choo
B_nam
usern
B_subca
Book_i
b_id
B_des
Cart_id
Bk_nam
Book
Adds
to
Cart
qty
B_publish
price
amoun
Editio
page
ISBN
Write
query
confir
query
Cart_i
orderID
email
order
contact
Order_
date
con_id
usern
amoun

+

## Chapter - 3

# Implement And Testing

# 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.

# Source Code

MONGODB_URL =
'mongodb+srv://shikha:shikha@cluster0.tivay.mongodb.net/myFirstDatabase?retryWrites=true&w=majority'

CLOUD_API_KEY = 452273692636939

CLOUD_API_SECRET = 0kZeA9uGffyq-_y2ny04aySiUAo

CLOUD_NAME = shikha

REFRESH_TOKEN_SECRET =
YOUR_REFRESH_TOKEN_SECRET

ACCESS_TOKEN_SECRET = YOUR_ACCESS_TOKEN_SECRET

```
import React, {useContext, useState, useEffect} from 'react'
```

```
import {useParams, Link} from 'react-router-dom'
import {GlobalState} from '../../../GlobalState'
import ProductItem from '../utils/productItem/ProductItem'


function DetailProduct() {
    const params = useParams()
    const state = useContext(GlobalState)
    const [products] = state.productsAPI.products
    const addCart = state.userAPI.addCart
    const [detailProduct, setDetailProduct] = useState([])

    useEffect(() =>{
        if(params.id){

            products.forEach(product => {
                if(product._id === params.id) setDetailProduct(product)
            })
        }
    },[params.id, products])

    if(detailProduct.length === 0) return null;

    return (
        <>
            <div className="detail">
                <img src={detailProduct.images.url} alt="" />
                <div className="box-detail">
                    <div className="row">
                        <h2>{detailProduct.title}</h2>
                        <h6>#id: {detailProduct.product_id}</h6>
                    </div>
                    <span>$ {detailProduct.price}</span>
                    <p>{detailProduct.description}</p>
                    <p>{detailProduct.content}</p>
                    <p>Sold: {detailProduct.sold}</p>
                    <Link to="/cart" className="cart"
                    onClick={() => addCart(detailProduct)}>
                        Buy Now
                    </Link>
                </div>
            </div>

            <div>
                <h2>Related products</h2>
                <div className="products">
                    {
                        products.map(product => {
                            return product.category === detailProduct.category
```

```
                                ? <ProductItem key={product._id} product={product} /> : null
                        })
                    }
                </div>
            </div>
        </>
    )
}

export default DetailProduct
```

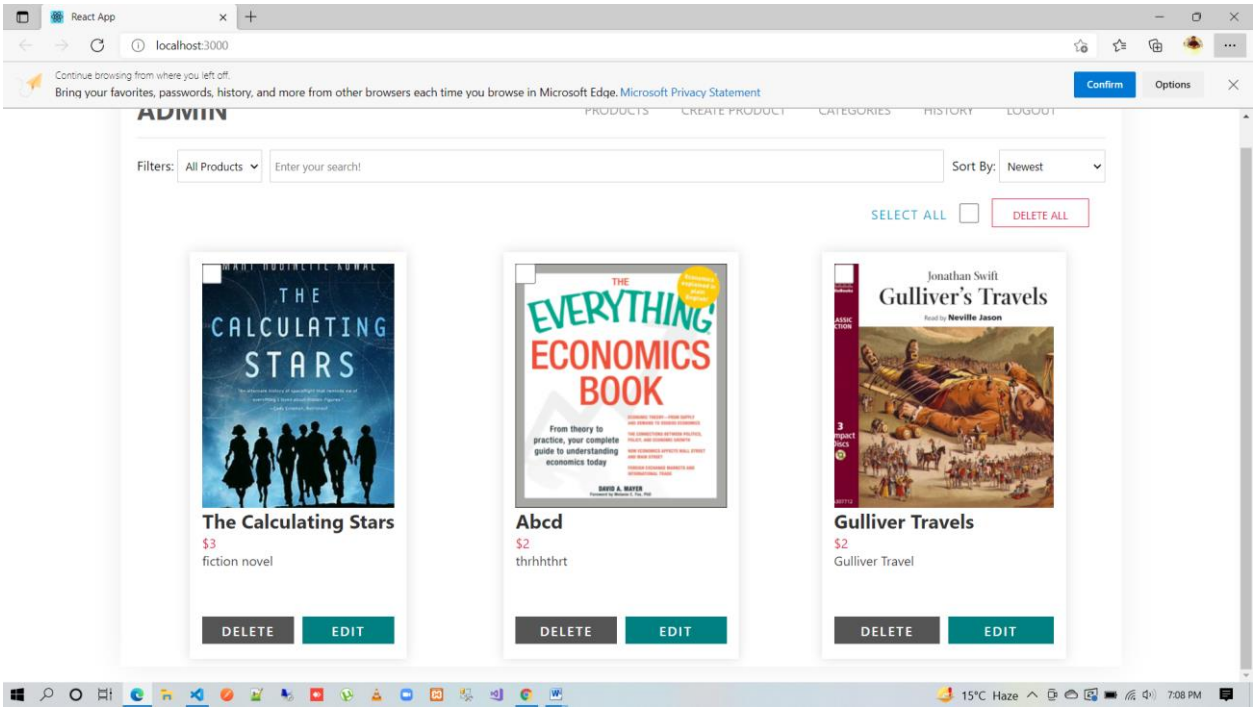
```
const Users = require('../models/userModel')

const authAdmin = async (req, res, next) =>{
    try {
        // Get user information by id
        const user = await Users.findOne({
            _id: req.user.id
        })
        if(user.role === 0)
            return res.status(400).json({msg: "Admin resources access denied"})

        next()

    } catch (err) {
        return res.status(500).json({msg: err.message})
    }
}

module.exports = authAdmin
```

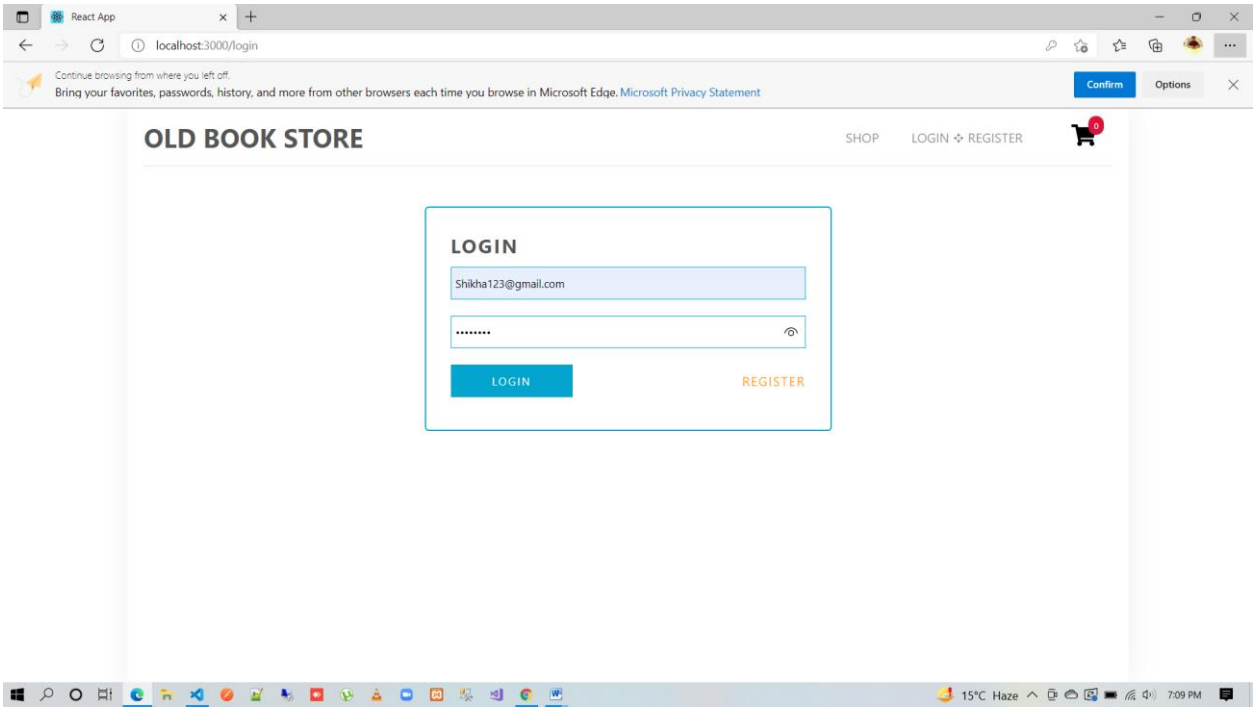
```
import React, {useContext, useState, useEffect} from 'react'
import {useParams, Link} from 'react-router-dom'
import {GlobalState} from '../../../GlobalState'
import ProductItem from '../utils/productItem/ProductItem'


function DetailProduct() {
    const params = useParams()
    const state = useContext(GlobalState)
    const [products] = state.productsAPI.products
    const addCart = state.userAPI.addCart
    const [detailProduct, setDetailProduct] = useState([])

    useEffect(() =>{
        if(params.id){

            products.forEach(product => {
                if(product._id === params.id) setDetailProduct(product)
            })
        }
    },[params.id, products])

    if(detailProduct.length === 0) return null;

    return (
        <>
            <div className="detail">
                <img src={detailProduct.images.url} alt="" />
                <div className="box-detail">
                    <div className="row">
                        <h2>{detailProduct.title}</h2>
                        <h6>#id: {detailProduct.product_id}</h6>
                    </div>
                    <span>$ {detailProduct.price}</span>
                    <p>{detailProduct.description}</p>
                    <p>{detailProduct.content}</p>
                    <p>Sold: {detailProduct.sold}</p>
                    <Link to="/cart" className="cart"
                    onClick={() => addCart(detailProduct)}>
                        Buy Now
                    </Link>
                </div>
            </div>

            <div>
                <h2>Related products</h2>
                <div className="products">
```

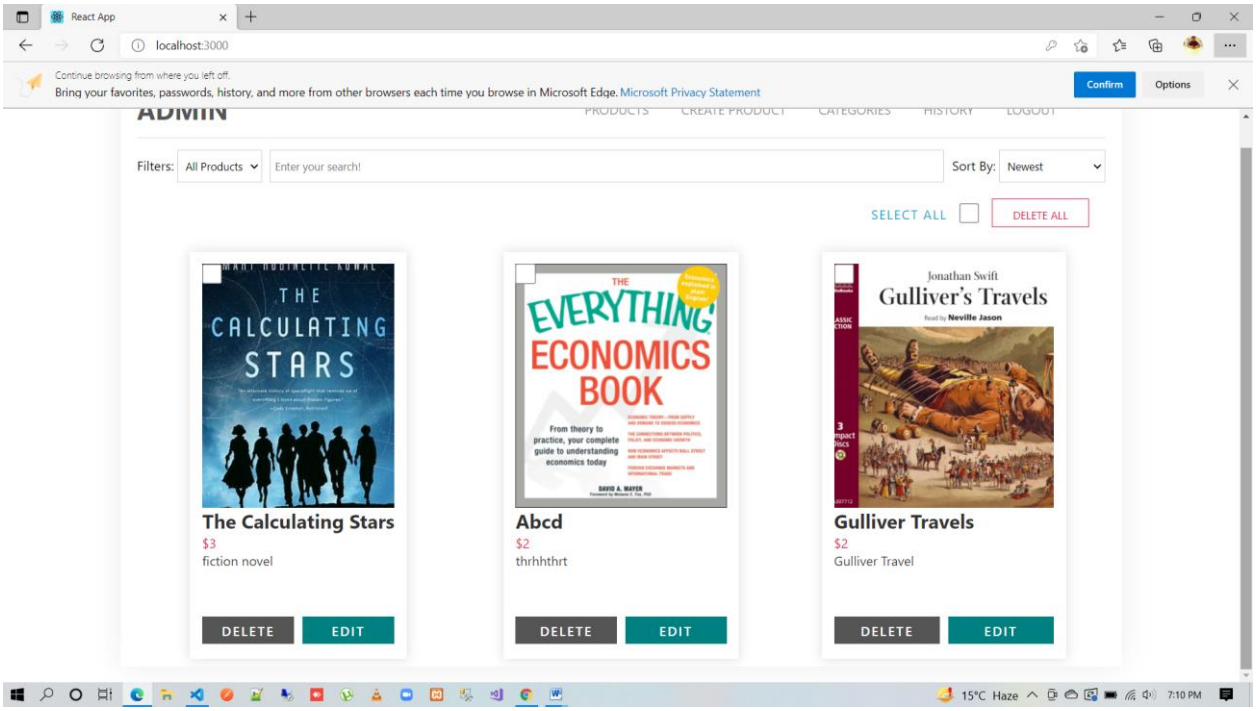
```
                    {
                        products.map(product => {
                            return product.category === detailProduct.category
                                ? <ProductItem key={product._id} product={product} /> : null
                        })
                    }
                </div>
            </div>
        </>
    )
}

export default DetailProduct
```

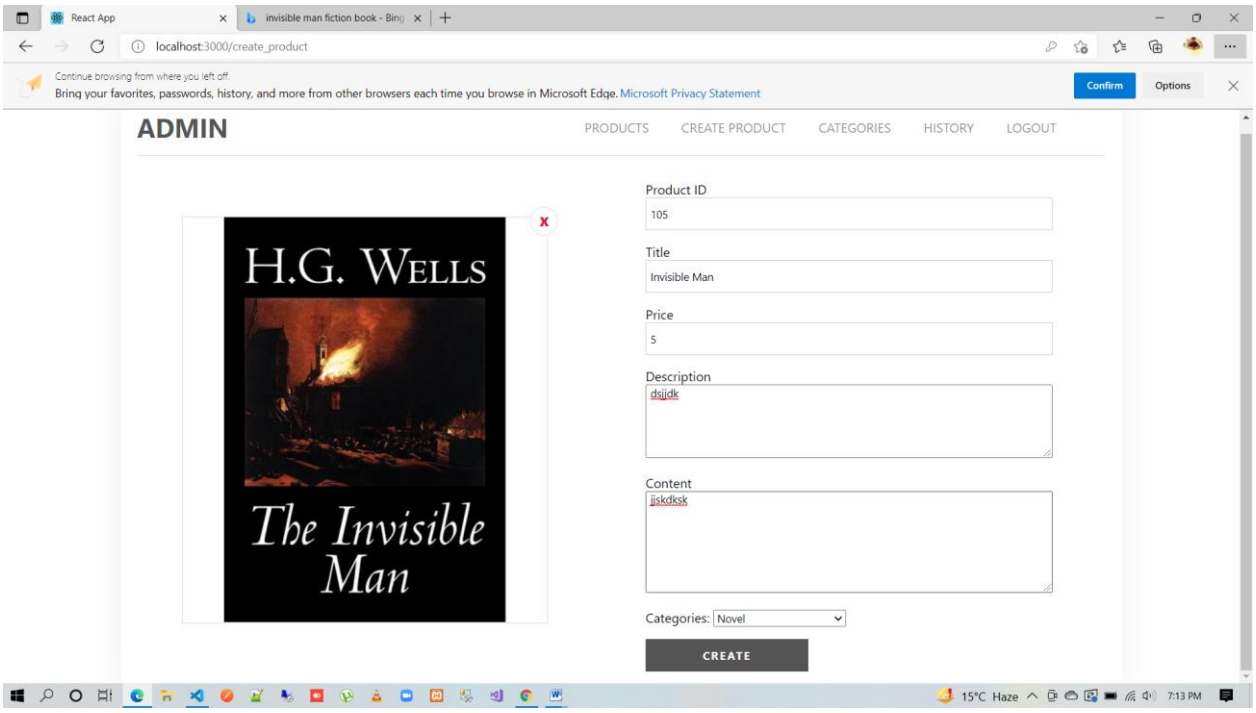
```
const Users = require('../models/userModel')

const authAdmin = async (req, res, next) =>{
    try {
        // Get user information by id
        const user = await Users.findOne({
            _id: req.user.id
        })
        if(user.role === 0)
            return res.status(400).json({msg: "Admin resources access denied"})

        next()

    } catch (err) {
        return res.status(500).json({msg: err.message})
    }
}

module.exports = authAdmin
```

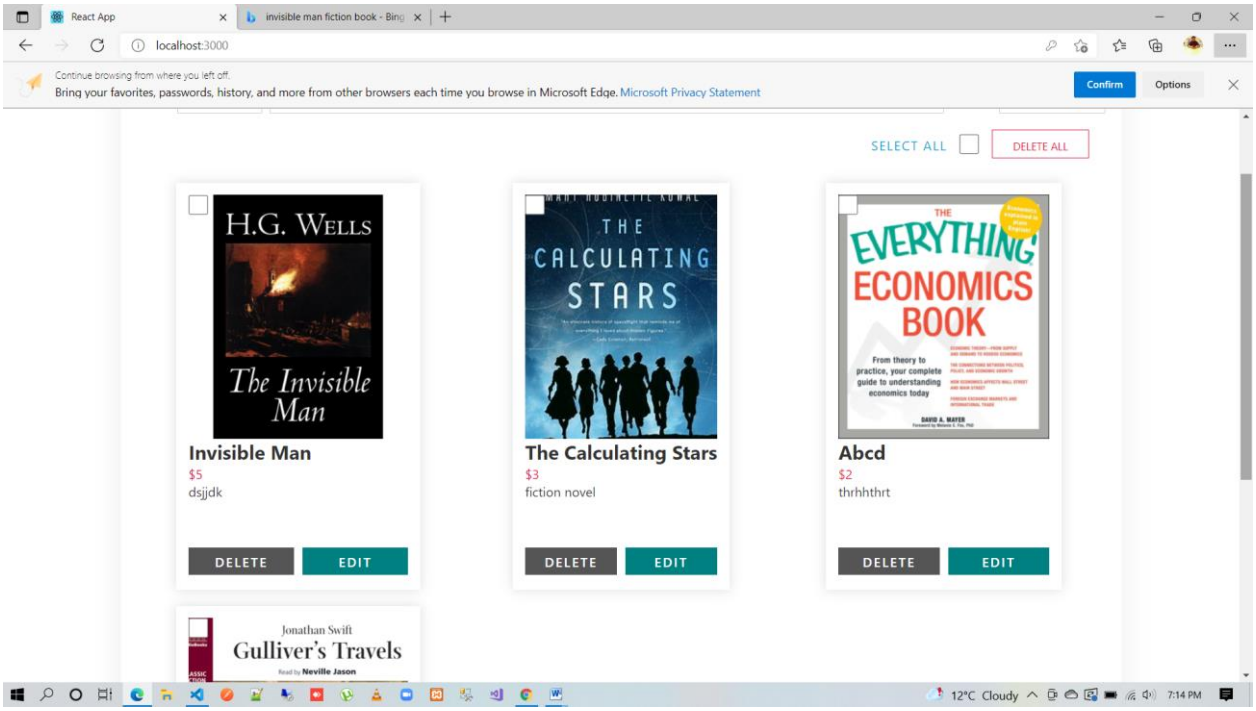
# Output

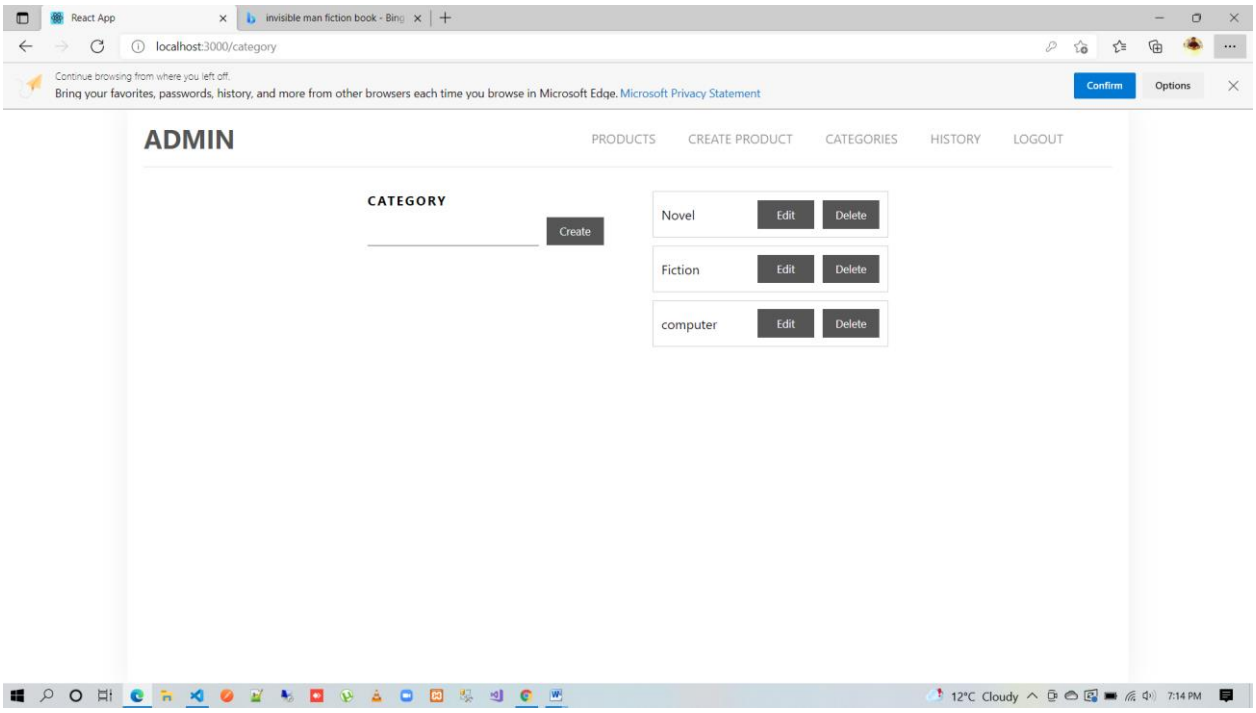

React App
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
ADMIN
PRODUCTS
CREATE PRODUCT
CATEGORIES
HISTORY
LOGOUT
Filters:
All Products
Enter your search!
Sort By:
Newest
SELECT ALL
DELETE ALL
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
DELETE
EDIT
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
DELETE
EDIT
Jonathan Swift
Gulliver's Travels
Gulliver Travels
$2
Gulliver Travel
DELETE
EDIT
15°C Haze
7:08 PM

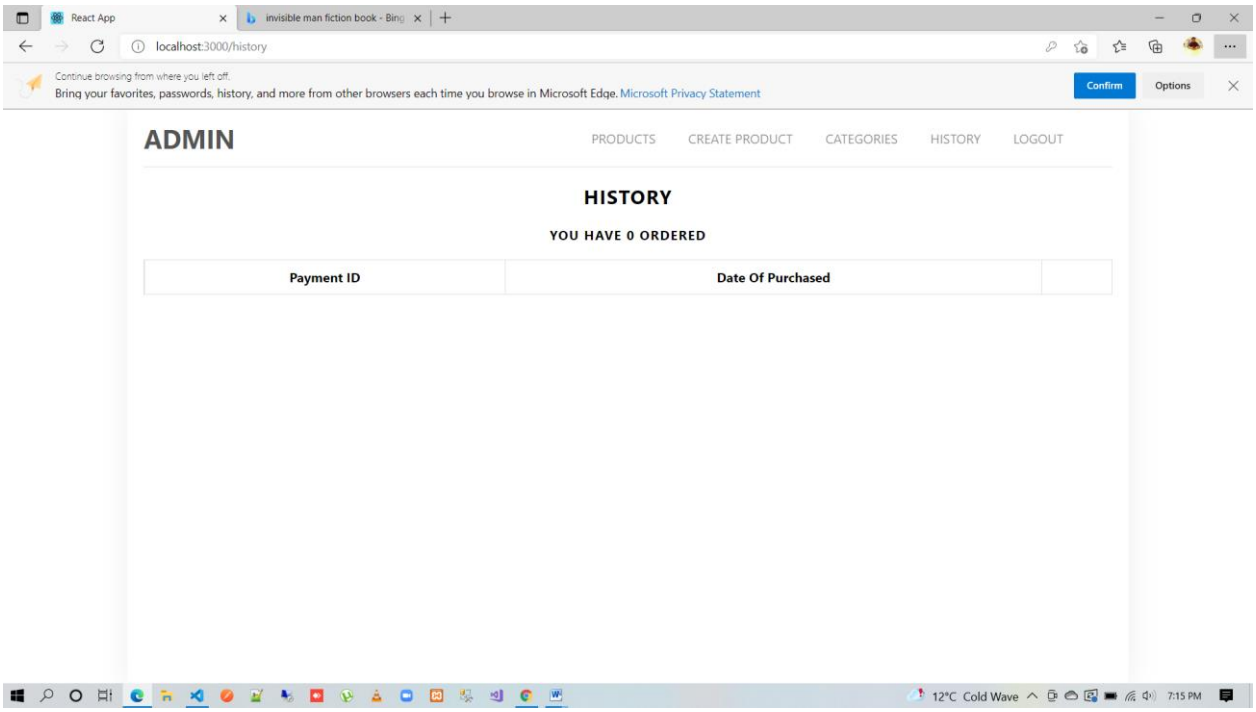

React App
localhost:3000/login
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
LOGIN
Shikha123@gmail.com
LOGIN
REGISTER
15°C Haze
7:09 PM

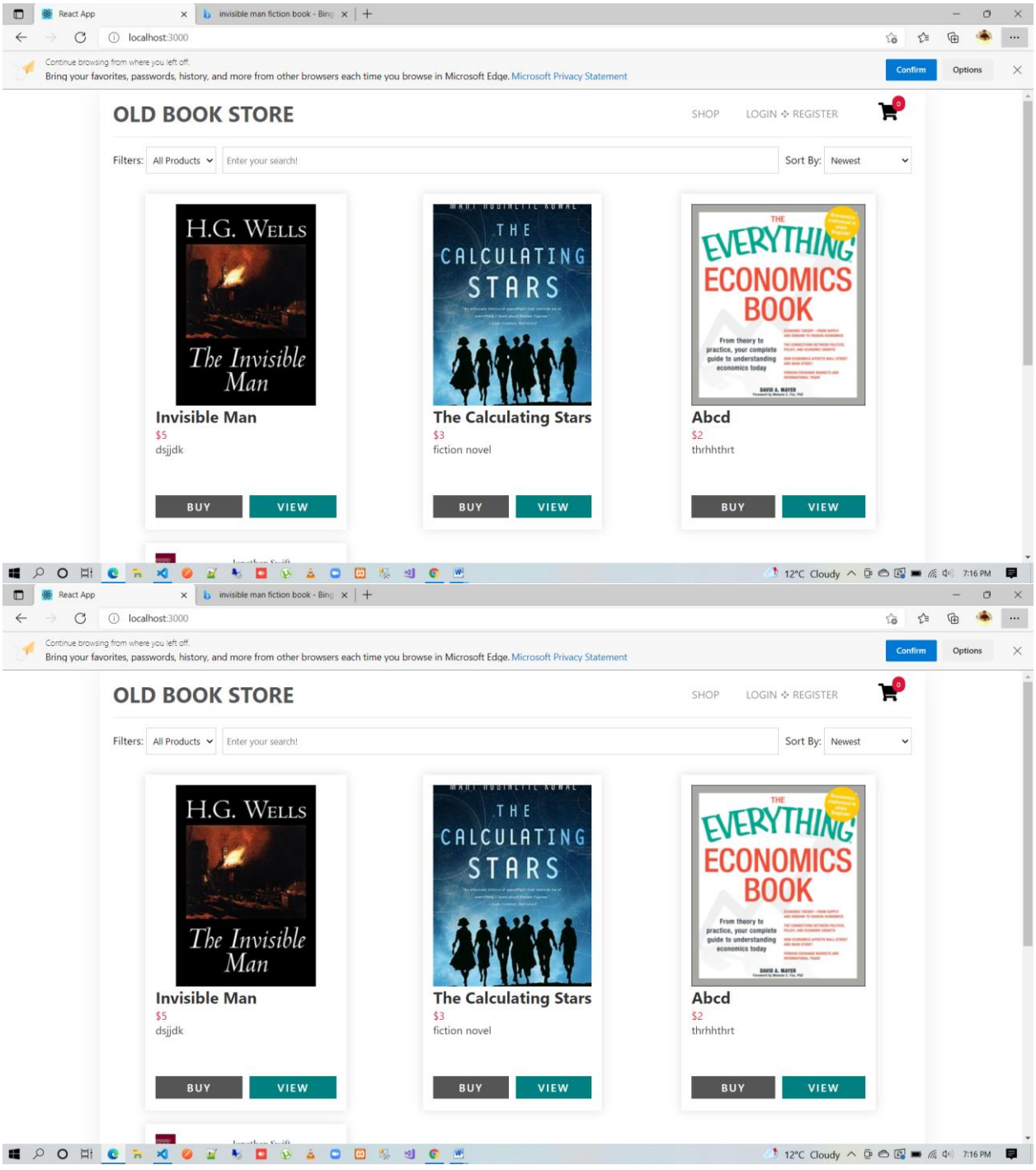

React App
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
ADMIN
PRODUCTS
CREATE PRODUCT
CATEGORIES
HISTORY
LOGOUT
Filters:
All Products
Enter your search!
Sort By:
Newest
SELECT ALL
DELETE ALL
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
DELETE
EDIT
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
DELETE
EDIT
Jonathan Swift
Gulliver's Travels
Read by Neville Jason
Gulliver Travels
$2
Gulliver Travel
DELETE
EDIT
15°C Haze
7:10 PM


React App
invisible man fiction book - Bing
localhost:3000/create_product
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
ADMIN
PRODUCTS
CREATE PRODUCT
CATEGORIES
HISTORY
LOGOUT
H.G. WELLS
The Invisible Man
Product ID
105
Title
Invisible Man
Price
5
Description
dsjidk
Content
jjskdksk
Categories: Novel
CREATE
15°C Haze
7:13 PM

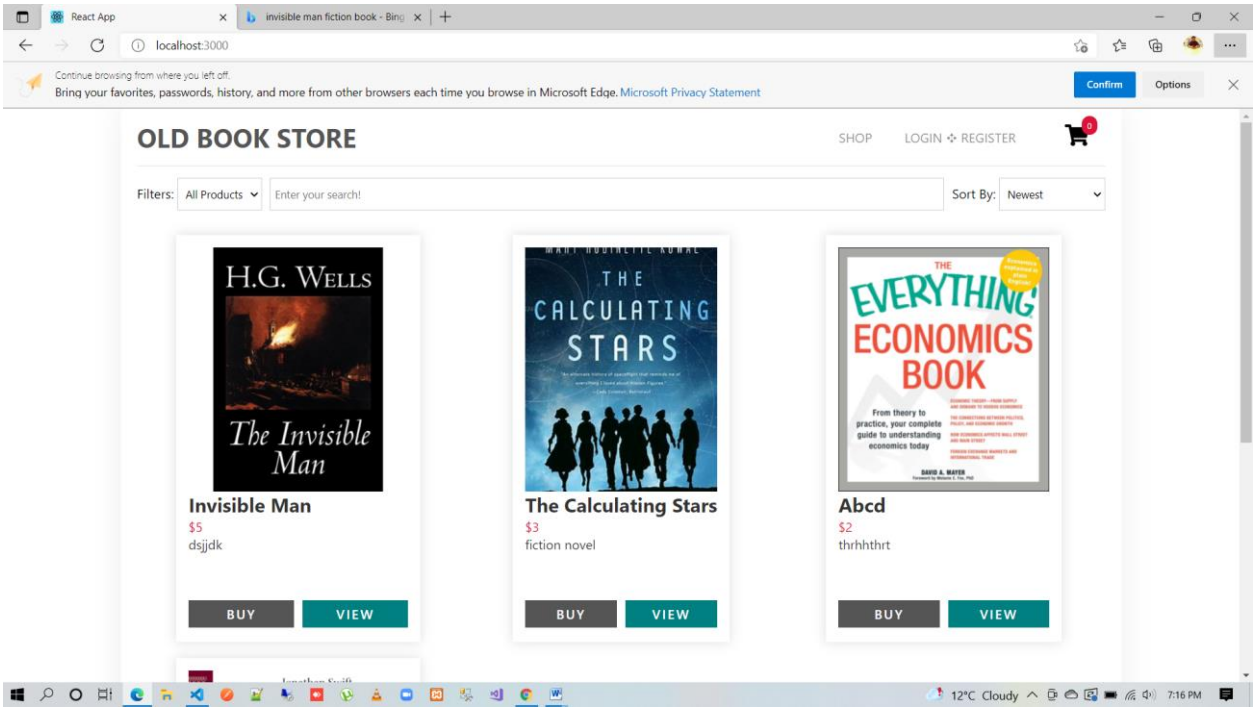

React App
invisible man fiction book - Bing
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
SELECT ALL
DELETE ALL
H.G. WELLS
The Invisible Man
Invisible Man
$5
dsjjdk
DELETE
EDIT
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
DELETE
EDIT
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
DELETE
EDIT
Jonathan Swift
Gulliver's Travels
12°C Cloudy
7:14 PM

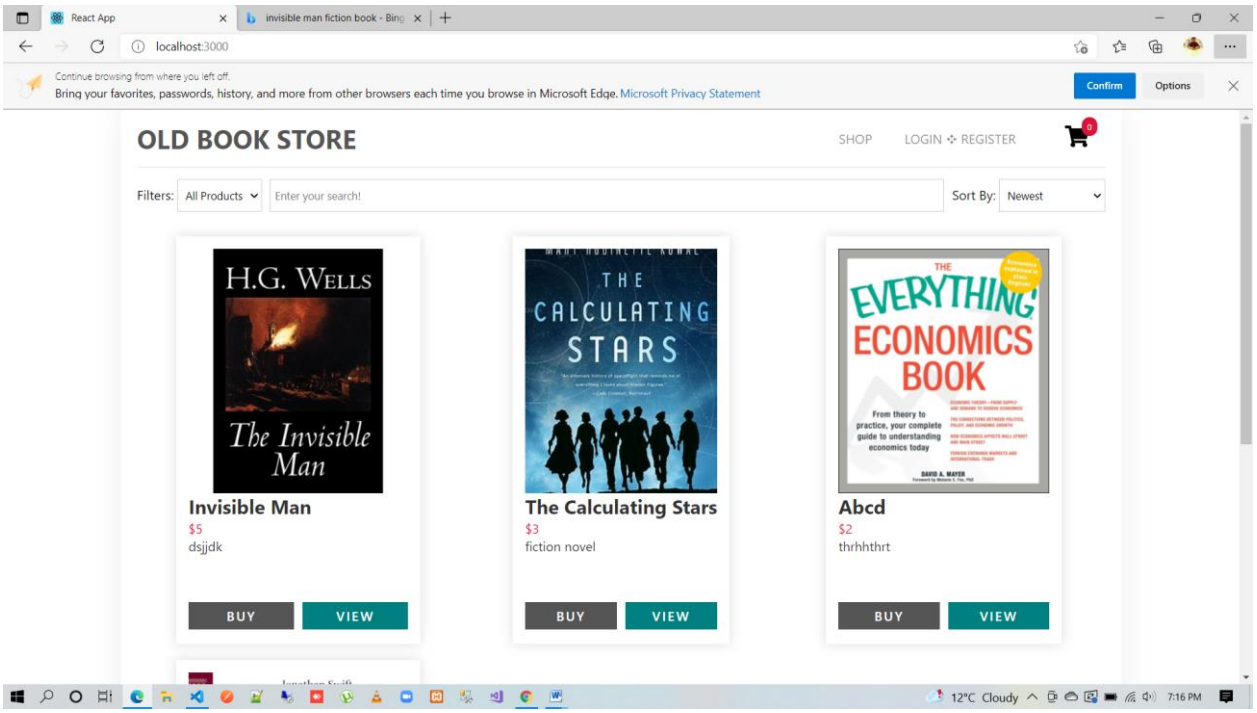

React App
invisible man fiction book - Bing
localhost:3000/category
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
ADMIN
PRODUCTS
CREATE PRODUCT
CATEGORIES
HISTORY
LOGOUT
CATEGORY
Create
Novel
Edit
Delete
Fiction
Edit
Delete
computer
Edit
Delete
12°C Cloudy
7:14 PM

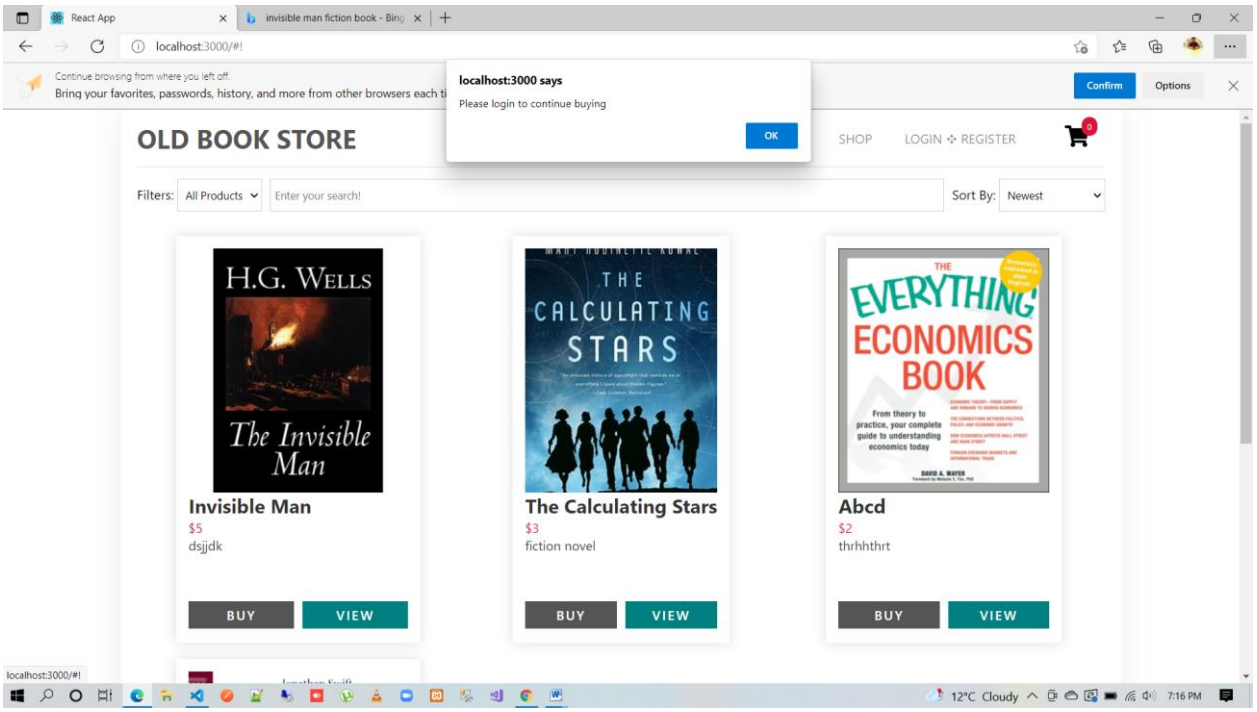

React App
invisible man fiction book - Bing
localhost:3000/history
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
ADMIN
PRODUCTS
CREATE PRODUCT
CATEGORIES
HISTORY
LOGOUT
HISTORY
YOU HAVE 0 ORDERED
Payment ID
Date Of Purchased
12°C Cold Wave
7:15 PM

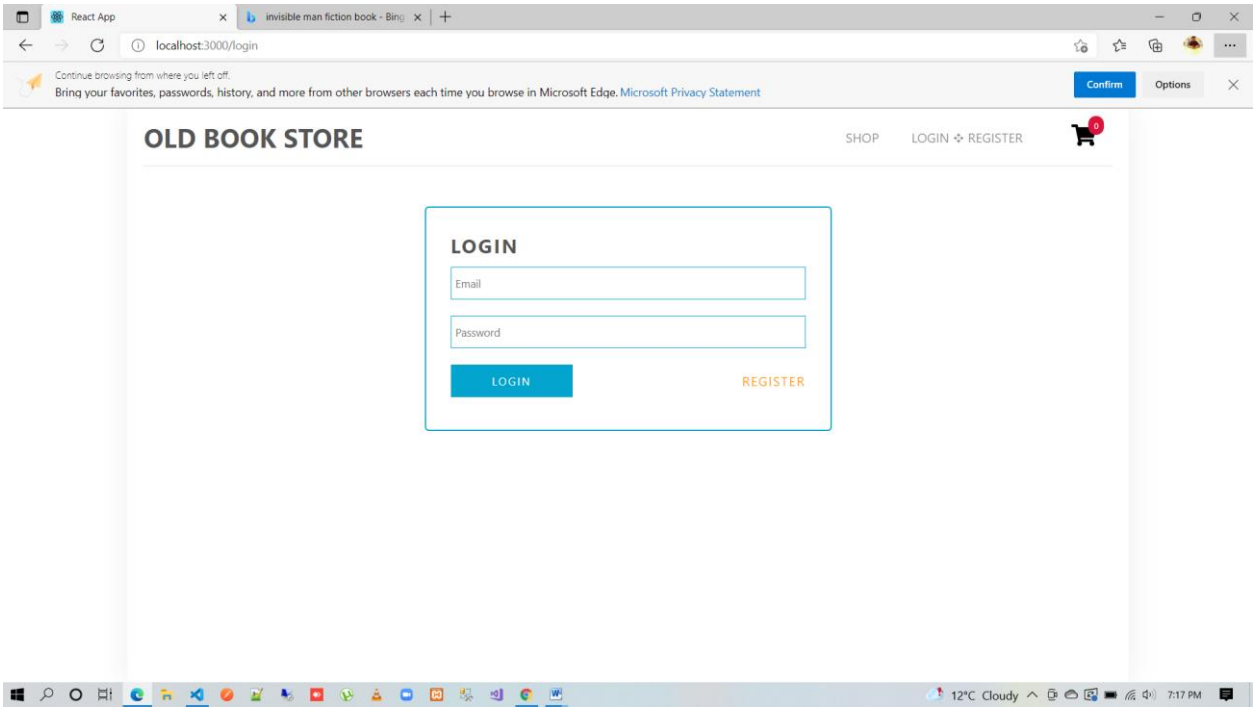



React App
invisible man fiction book - Bing
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
Filters:
All Products
Enter your search!
Sort By:
Newest
H.G. WELLS
The Invisible Man
Invisible Man
$5
dsjjdk
BUY
VIEW
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
BUY
VIEW
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
BUY
VIEW
12°C Cloudy
7:16 PM


React App
invisible man fiction book - Bing
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
Filters:
All Products
Enter your search!
Sort By:
Newest
H.G. Wells
The Invisible Man
Invisible Man
$5
dsjjdk
BUY
VIEW
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
BUY
VIEW
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
BUY
VIEW
12°C Cloudy
7:16 PM


React App
invisible man fiction book - Bing
localhost:3000
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
Filters:
All Products
Enter your search!
Sort By:
Newest
H.G. Wells
The Invisible Man
Invisible Man
$5
dsjjdk
BUY
VIEW
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
BUY
VIEW
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
BUY
VIEW
12°C Cloudy
7:16 PM


React App
invisible man fiction book - Bing
localhost:3000/#!
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each ti
Confirm
Options
localhost:3000 says
Please login to continue buying
OK
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
Filters:
All Products
Enter your search!
Sort By:
Newest
H.G. WELLS
The Invisible Man
Invisible Man
$5
dsjjdk
BUY
VIEW
THE CALCULATING STARS
The Calculating Stars
$3
fiction novel
BUY
VIEW
THE EVERYTHING ECONOMICS BOOK
From theory to practice, your complete guide to understanding economics today
Abcd
$2
thrhhthrt
BUY
VIEW
localhost:3000/#!
12°C Cloudy
7:16 PM


React App
invisible man fiction book - Bing
localhost:3000/login
Continue browsing from where you left off.
Bring your favorites, passwords, history, and more from other browsers each time you browse in Microsoft Edge. Microsoft Privacy Statement
Confirm
Options
OLD BOOK STORE
SHOP
LOGIN ✣ REGISTER
LOGIN
Email
Password
LOGIN
REGISTER
12°C Cloudy
7:17 PM

# Bibliography

I have done this project with the help of our supervisor **Mr. Naresh Chandra, Assistant Professor, KIET Group of Institutions.**
And taking references from the following:
https://www.w3schools.com/html/
https://www.w3schools.com/css/
https://www.w3schools.com/js/

## I used:

- VS Code
- MERN