VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report

on

ONLINE BOOKSTORE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

RITESH AGARWAL
1RN18IS087

Under the Guidance of

Mr. R Rajkumar
Associate Professor
Department of ISE

Department of Information Science and Engineering
RNS Institute of Technology
Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

2021 – 2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Online Bookstore Management System*** has been successfully completed by **Ritesh Agarwal (1RN18IS087)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mr. R Rajkumar	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Associate Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

I, **RITESH AGARWAL [USN:1RN18IS087]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship Work entitled ***Online Bookstore Management System has*** been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

ABSTRACT

Nowadays, the network plays an import role in people's life. In the process of the improvement of the people's living standard, people's demands of the life's quality and efficiency is higher, the traditional bookstore's inconvenience gradually emerge, and the online bookstore has gradually been used in public. The online bookstore is a revolution of book industry. The traditional bookstores' operation time, address and space is limited, so the types of books and books to find received a degree of restriction. But the online bookstore broke the management mode of traditional bookstore, as long as you have a computer, you can buy the book anywhere, saving time and effort, shortening the time of book selection link effectively. The online bookstore system based on the principle of provides convenience and service to people.

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mrs. Kavya TC** , Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Ramesh Nediyadath, Designation, TechieAid**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

RITESH AGARWAL

USN: 1RN18IS087

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CSS	-	Cascading style sheets
DBMS	-	Database Management System
ER	-	Entity Relationship
HTML	-	Hypertext Markup Language
HTTP	-	Hypertext Transfer Protocol
JS	-	JavaScript
PHP	-	PHP Hypertext Preprocessor
SQL	-	Structured Query Language

Chapter 1

INTRODUCTION

1.1 Background

A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, the database itself to capture and analyze the data and provides facilities to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.2 Introduction to Online Bookstore Management System

The main objective of the project is to create an online book store that allows users to purchase a book based on title, author and category. The selected books are displayed in a tabular format and the user can order their books online by using cash on delivery payment method. Using this application, the user can purchase a book online instead of going out to a book store and wasting time.

There are many online book stores like Powell's, Amazon which were designed using Html. I wanted to develop a similar application like that.

Online Book store is an online web application where the customer can purchase books online.

Through this application the customers can find the book which he/she desired to buy, later can add to the shopping cart and finally purchase using cash on delivery payment method. The user can login using his account details or new customers can set up an account very quickly. They can easily register in the bookstore with their email and by creating a suitable password. The books are divided into many categories like best sellers, business and management, fantasy, horror etc.

The Administrator will have additional functionalities when compared to the common user. He can view the registered users, can add, update and delete books which are present in the bookstore and can view the orders which are placed by the customers.

Chapter 2

E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

2.1 Description of ER Diagram

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

2.1.1 Components of Airline Reservation System, E-R Diagram

Entity types like **ADMIN** and **CART** are in rectangular boxes.

1. Relationships like **REGISTERS** and **HAS** are in diamond boxes, attached to entity types with straight lines.
2. Attributes are shown in ovals like **Name** and **phone _no** each attached by a straight line to entity or relationship type.
3. Key attributes like **c_email** and **book_id** are underlined.

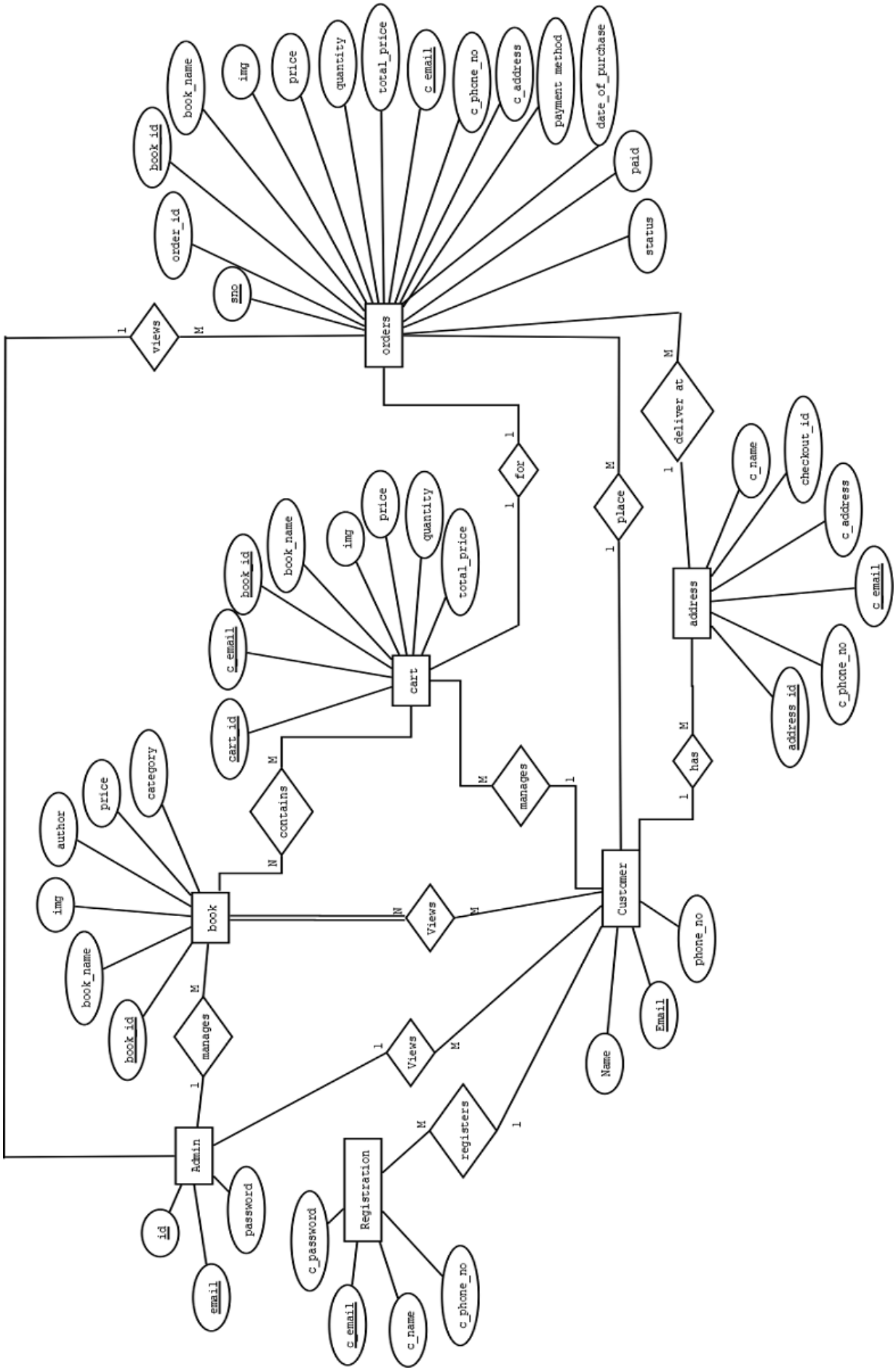

Figure 2.1 ER Diagram for Online Bookstore Management System

2.1.2 E-R Diagram Relationships Description

1. **CUSTOMER:REGISTRATION** is of cardinality 1:N as one customer can possibly have more than one email and therefore connected by an **IN** relationship.
2. **ADMIN:CUSTOMER** is of cardinality 1:N as one admin can view more than one customer and therefore connected by an **IN** relationship.
3. **ADMIN:BOOK** is of cardinality 1:N as one admin can be manage multiple books, but all the books might not be managed by one admin..
4. **CUSTOMER:BOOK** is of cardinality M:N as one customer can view multiple books and one book can be viewed by more than one customer and therefore connected by **VIEW** relationship. There is total participation from BOOK as all books must be viewed by some customer, but partial participation from CUSTOMER as a customer may or may not view a book.
5. **CUSTOMER:ADDRESS** is of cardinality of 1:N as one customer can have multiple addresses.
6. **CART:BOOK** is of cardinality of 1:N as one shopping cart can contain multiple books.
7. **CUSTOMER:CART** is of cardinality of 1:N as one customer can add/delete multiple books in the shopping cart.
8. **CART:ORDER** is of cardinality of 1:1 as a customer can place one order for one shopping cart and therefore connected by PLACE.
9. **CUSTOMER:ORDER** is of cardinality of 1:N as one customer can place more than one order.
10. **ADDRESS:ORDER** is of cardinality of 1:N as more than one order can be delivered at one address but one order can’t be delivered at multiple addresses.

2.2 Description of Relational Schema Diagram

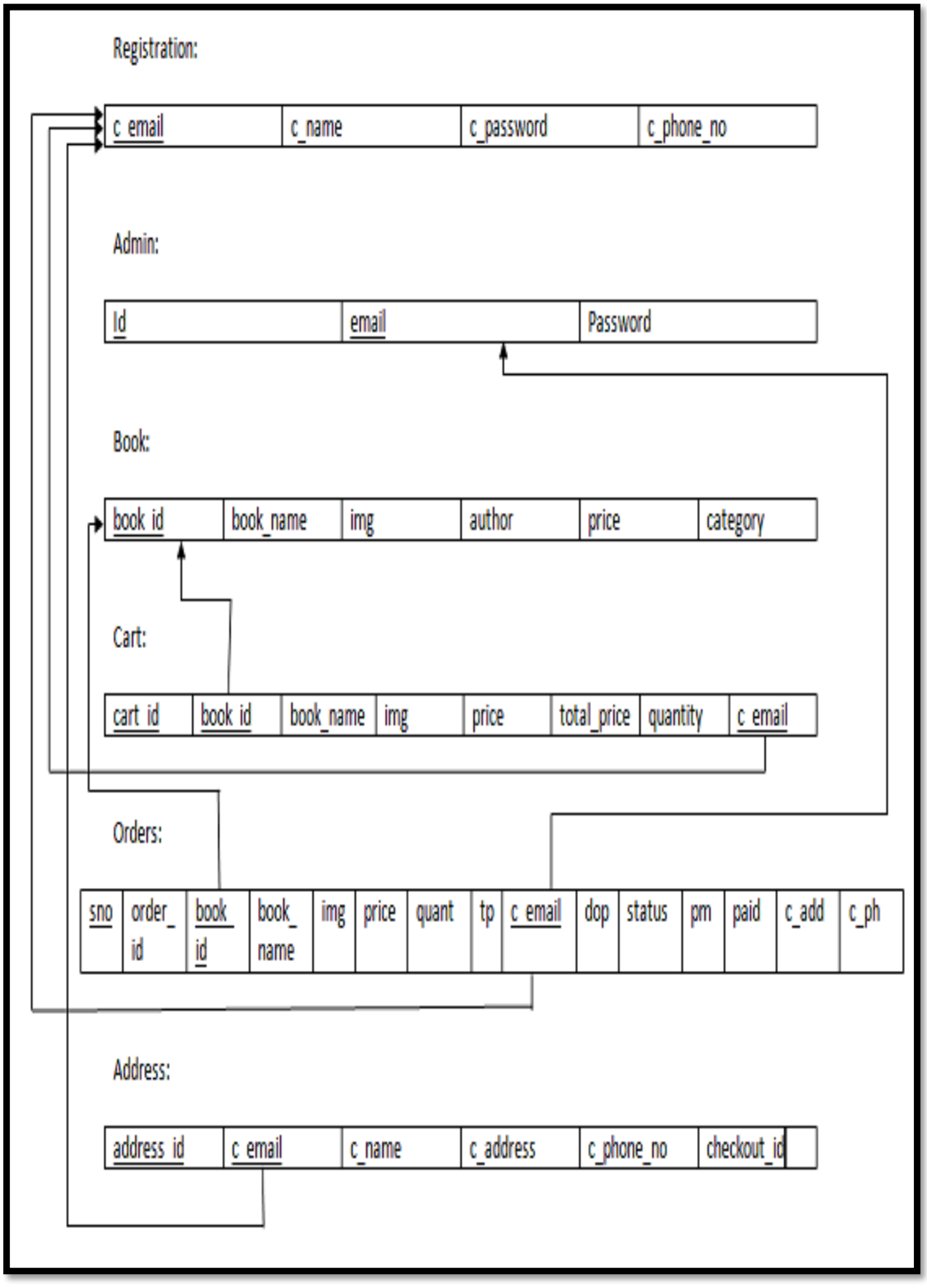

Figure 2.2 Relational Schema Diagram for Online Bookstore Management System

2.2.1 General Constraints

1. **NULL Constraint**: Attributes that are under NOT NULL constraints have to be filled compulsorily. Almost all the attributes in the project are under NOT NULL constraint.
2. **Entity Integrity Constraint**: This constraint makes sure that no primary key can have a NULL value assigned to it. The primary keys involved in the project include:
    - c_email
    - book_id
    - address_id
    - sno
    - cart_id
    - email
3. **Referential Integrity Constraints**: A table in the back end of the project may have references pointing to an attribute in another table. The various tables are also linked with multiple foreign keys which are all set to cascade any update or delete operation on the attribute in the main table. The various Foreign Key attributes are:
    - c_email
    - book_id

2.2.2 Schema Description

The above Figure.2.2 shows the relational schema of Airline Reservation System. It has the following entities.

1. **REGISTRATION:** This table contains the details like customer email, name, password, phone number. c_email is the primary key.
2. **ADMIN:** This table contains the login details of the administrator. It has 3 columns id, email, password.
3. **BOOK:** This table consists the list of books available in the application. It has 6 columns book_id, book_name, img, author, price, category.
4. **CART:** This table consists of all the books that the customer wants to buy. It has 8 columns namely cart_id, book_id, book_name, img, price, total_price, quantity, c_email.
5. **ORDER:** This table consists of a list of all the orders made by the customer registered with the system. It has the following columns, sno, order_id, book_id,

book_name, img, price, quantity, total_price, date_of_purchase, status, payment_method, paid, c_email, c_address, c_phone_no. An assumption is made that, all flights taken into consideration are direct flights from source to destination.

6. **ADDRESS:** This table contains a list of addresses on which the customer wants the order to be delivered. It has address_id, c_email, c_name, c_address, c_phone_no, checkout_id.

Chapter 3

SYSTEM DESIGN

3.1 Table Description

This section describes the tables that are associated with this application. There are total six tables in the database namely registration which contains the information about customer, book which contains the book details, admin table contains the information of the admin, cart contains the books selected by the customer, orders table contains the order placed by the customer and address table contains the delivery information of the customer

REGISTRATION

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	c_email	varchar(50)	utf8mb4_general_ci		No	*None*		
2	c_name	varchar(50)	utf8mb4_general_ci		No	*None*		
3	c_password	varchar(200)	utf8mb4_general_ci		No	*None*		
4	c_phone_no	varchar(11)	utf8mb4_general_ci		No	*None*		

Table 3.1 registration

BOOK

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	book_id	int(11)			No	*None*		AUTO_INCREMENT
2	book_name	varchar(50)	utf8mb4_general_ci		No	*None*		
3	img	varchar(100)	utf8mb4_general_ci		No	*None*		
4	author	varchar(50)	utf8mb4_general_ci		No	*None*		
5	price	varchar(50)	utf8mb4_general_ci		No	*None*		
6	category	varchar(50)	utf8mb4_general_ci		No	*None*		

Table 3.2 Book

ADMIN

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	email	varchar(50)	utf8mb4_general_ci		No	*None*		
3	password	varchar(100)	utf8mb4_general_ci		No	*None*		

Table 3.3 Admin

CART

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	cart_id	int(25)			No	None		AUTO_INCREMENT
2	book_id	int(11)			No	None		
3	book_name	varchar(50)	utf8mb4_general_ci		No	None		
4	img	varchar(100)	utf8mb4_general_ci		No	None		
5	price	varchar(50)	utf8mb4_general_ci		No	None		
6	total_price	varchar(50)	utf8mb4_general_ci		No	None		
7	quantity	varchar(50)	utf8mb4_general_ci		No	None		
8	c_email	varchar(50)	utf8mb4_general_ci		No	None		

Table 3.5 Cart

ORDERS

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	sno	int(11)			No	None		AUTO_INCREMENT
2	order_id	varchar(200)	utf8mb4_general_ci		No	None		
3	book_id	int(11)			No	None		
4	book_name	varchar(50)	utf8mb4_general_ci		No	None		
5	img	varchar(100)	utf8mb4_general_ci		No	None		
6	price	varchar(50)	utf8mb4_general_ci		No	None		
7	quantity	varchar(50)	utf8mb4_general_ci		No	None		
8	total_price	varchar(50)	utf8mb4_general_ci		No	None		
9	c_email	varchar(50)	utf8mb4_general_ci		No	None		
10	date_of_purchase	varchar(50)	utf8mb4_general_ci		No	None		
11	status	varchar(50)	utf8mb4_general_ci		No	None		
12	payment_method	varchar(50)	utf8mb4_general_ci		No	None		
13	paid	varchar(50)	utf8mb4_general_ci		No	None		
14	c_address	varchar(100)	utf8mb4_general_ci		No	None		
15	c_phone_no	varchar(11)	utf8mb4_general_ci		No	None		

Table 3.6 books

ADDRESS

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	address_id	int(11)			No	None		AUTO_INCREMENT
2	c_email	varchar(50)	utf8mb4_general_ci		No	None		
3	c_name	varchar(50)	utf8mb4_general_ci		No	None		
4	c_address	varchar(100)	utf8mb4_general_ci		No	None		
5	c_phone_no	varchar(11)	utf8mb4_general_ci		No	None		
6	checkout_id	varchar(200)	utf8mb4_general_ci		No	None		

Table 3.7 address

3.2 Stored Procedure

One stored procedure is used when the admin wants to view all customers who are registered with the application.

3.2.1 Creating stored procedure getReagistration():

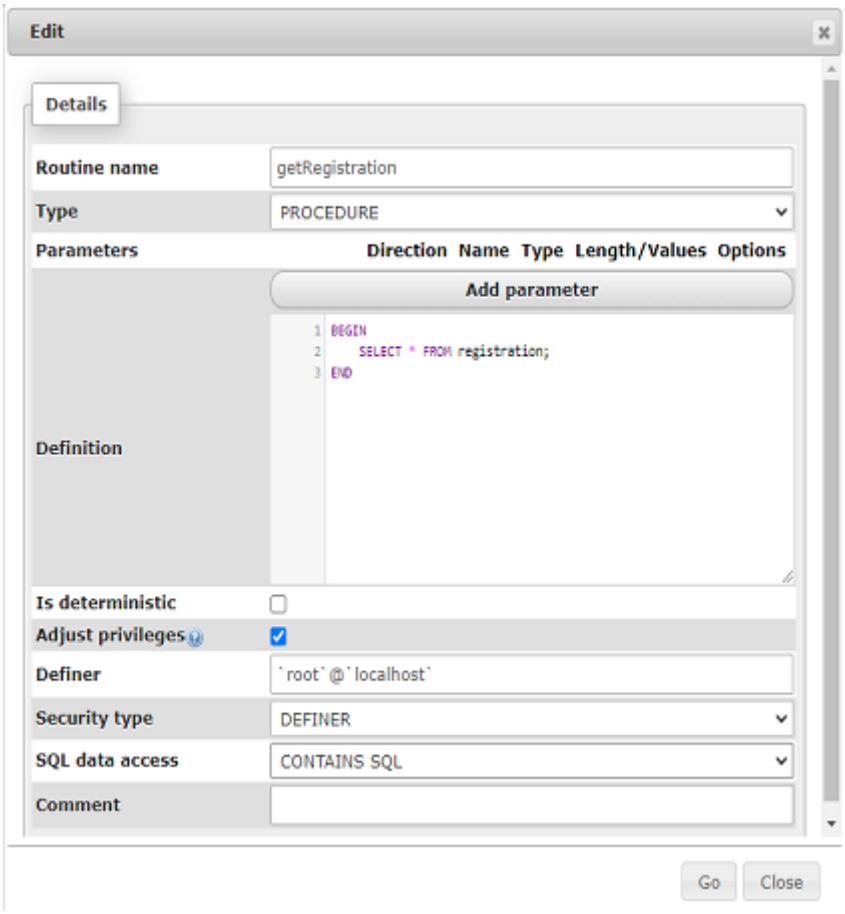

Figure 3.1 creating stored procedure getRegistration()

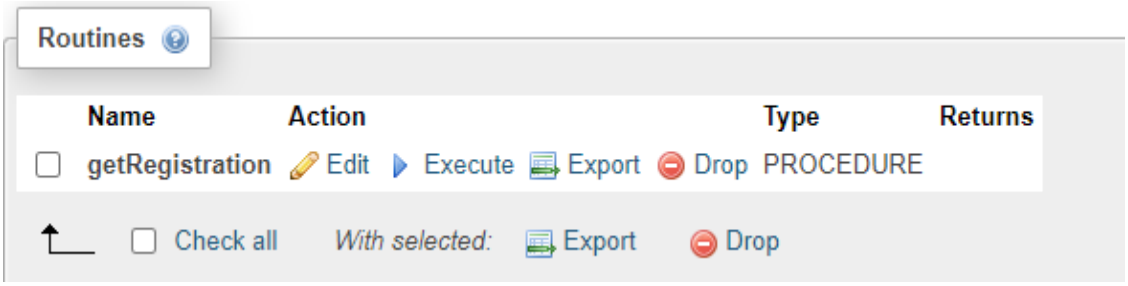

Figure 3.2 stored procedure getRegistration()

3.2.2 Calling stored procedure getReagistration():

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
```

```
$output ="";
if (isset($_POST['all'])) {
  $sql="CALL `getRegistration`();";
  $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0)
  {
    $output .= '<table class="table">
      <tr>
        <th>Username</th>
        <th>Name</th>
        <th>Mobile</th>
      </tr>';
    while ($row = mysqli_fetch_array($result)) {

      $output .= '
        <tr>
          <td>'.$row['c_email'].'</td>
          <td>'.$row['c_name'].'</td>
          <td>'.$row['c_phone_no'].'</td>
        </tr>';
    }
    $output .='</table>';

  }else {
    $output = '<div class="alert alert-danger">No record found</div>';
  }

}
function test_input($data){
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
 ?>
```

3.3 Trigger

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

In this project a trigger is added to impose a 10 % charge on price of each book.

3.3.1 Creating a trigger inc_price:

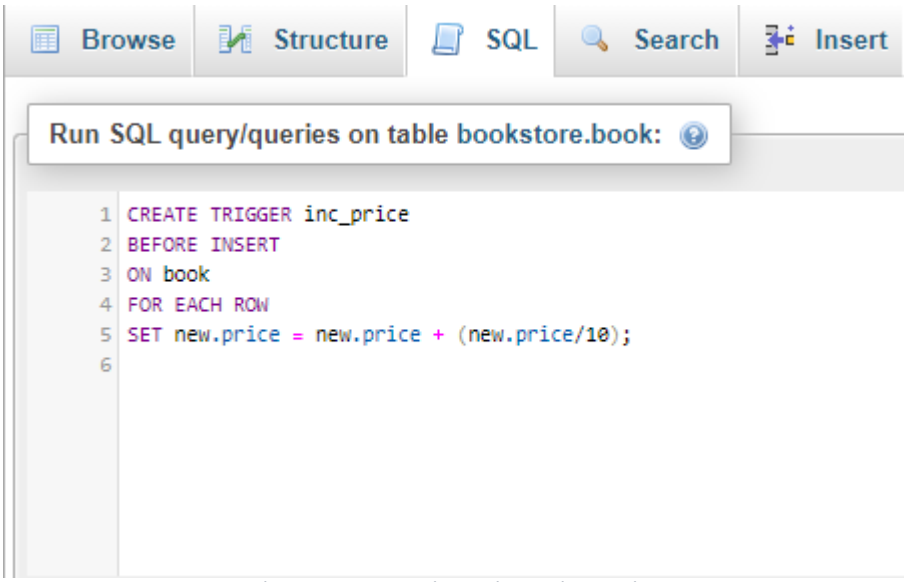

Figure 3.3 creating trigger inc_price

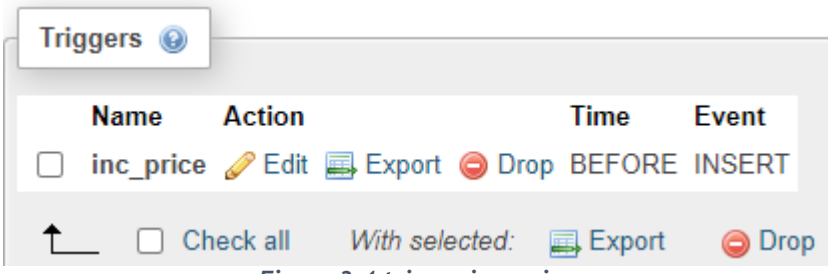

Figure 3.4 trigger inc_price

Chapter 4

IMPLEMENTATION

4.1 Front-end Development

The front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). Front-end developers design and construct the user experience elements on the web page or app including buttons, menus, pages, links, graphics and more.

4.1.1 Hypertext Markup Language

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. HTML is the standard markup language for creating Web pages. It stands for Hyper Text Markup Language. It describes the structure of a Web page. It consists of a series of elements. It elements tell the browser how to display the content. It elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

4.1.2 Cascading style sheets

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications. Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. CSS solved that issue. CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file. CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

4.1.3 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly. Advantages are: **Less server interaction, immediate feedback to the visitors, increased interactivity** and **richer interfaces.**

4.2 Back-end Development

Backend is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

4.2.1 Backend scripting language - PHP Hypertext Preprocessor

PHP is used as the server side scripting language. PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. It is compatible with all servers used today. It is easy to use and runs efficiently on the server side. It can run on various platforms like windows, Linux, UNIX, Mac OS-X etc. and since it is a scripting language, it comes with predefined functions which makes it easy to implement any logic necessary.

4.2.2 Web Server – Apache

Apache is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation. It allows website owners to serve content on the web — hence the name "web server". Although we call Apache a web server, it is not a physical server, but rather a software that runs on a server. Its job is to establish a connection between a server and the browsers of website visitors (Firefox, Google Chrome, Safari, etc.) while delivering files back and forth between them (client-server structure). Apache is a cross-platform software, therefore it works on both UNIX and Windows servers.

4.2.3 Database – MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. It is released under an open-source license. So you have nothing to pay to use it. It is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. It uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. It is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

4.3 User Flow Diagram

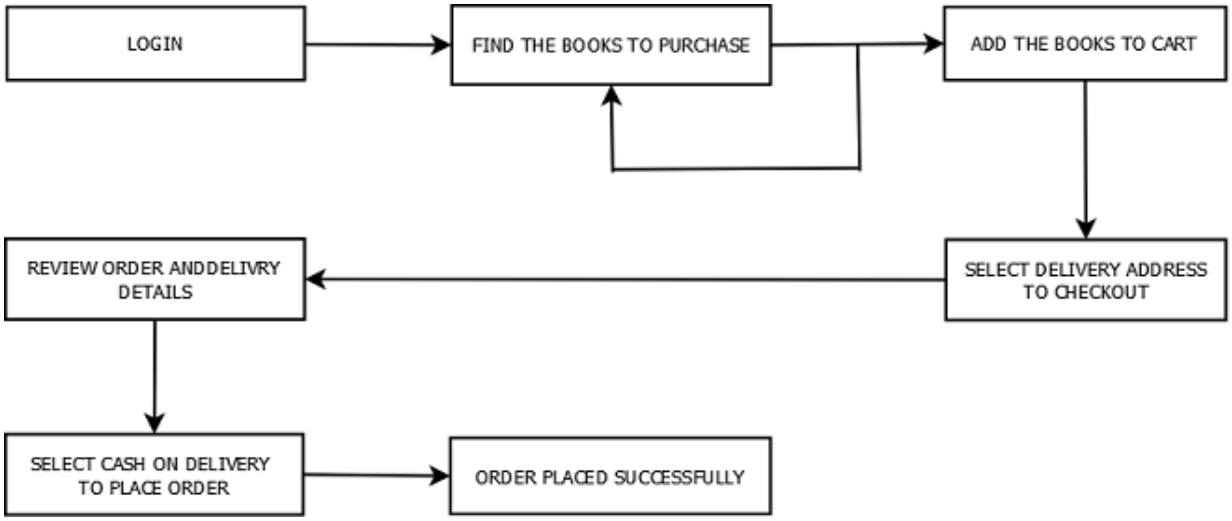

Figure 4.1 Order Successful user flow diagram

The above Figure 4.1 shows the user flow diagram for a successfully placed order. The customer logs in, selects the books he/she wants to buy and add those books to cart. The customer then selects the delivery address where he wants the order to be delivered. Then he/she reviews the order details before selecting the payment mode. After selecting the payment mode which is cash on delivery as for now the order gets successfully placed

4.4 Discussion of Code Segment

This section talks about the important code sections and modules that are implemented in the Airline Reservation System project. These modules add logic to the complete system, and make it function the way it is supposed to. It also talks about the integration between the front end HTML code and the back end MySQL database.

4.1.1 User

4.4.1.1 Login code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
$username=$_POST['username'];
$password = $_POST['password'];
if(isset($username) && isset($password))
{
 $sql = "SELECT * FROM registration WHERE c_email = '$username'";
 $result = mysqli_query($link,$sql);
```

```
 if (mysqli_num_rows($result)>0) {
  while ($row = mysqli_fetch_array($result)) {
   if (password_verify($password,$row['c_password'])) {
    $_SESSION['loggedin'] = true;
    //$_SESSION['id'] = $row['id'];
    $_SESSION['id'] = $row['c_email'];
    $_SESSION['name'] = $row['c_name'];
    echo 1;
   }else {
    echo 0;
   }
  }

 }else {
  echo 0;
 }
}
 ?>
```

4.4.1.2 Signup code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
$name = $_POST['name'];
$email = $_POST['email'];
$mobile = $_POST['mobile'];
$password = $_POST['password'];
if (empty($name) || empty($email) || empty($mobile) || empty($password)) {
 echo '<div class="alert alert-danger">Error occured, try again</div>';
}else {
 $sql = "INSERT INTO registration (c_email,c_name,c_password,c_phone_no) VALUES
(?,?,?,?)";
 $stmt = mysqli_prepare($link,$sql);
 mysqli_stmt_bind_param($stmt,'ssss',$param_username,$param_name,$param_password,$pa
ram_mobile);
```

```
  $param_username = $email;
  $param_name = $name;
  $param_password = password_hash($password,PASSWORD_DEFAULT);
  $param_mobile = $mobile;
  if (mysqli_stmt_execute($stmt))
  {
     echo '<div class="alert alert-success">Signup Successful</div>';

  }else {
     echo '<div class="alert alert-danger">Error Occured</div>';
  }
}
?>
```

4.4.1.3 Add to cart code:

```
<?php
session_start();
$link = mysqli_connect("localhost","root","","bookstore");
$status = "";
$var = $_POST['action'];
switch ($var) {
  case 'add_to_cart':
    $book_id = $_POST['book_id'];
    $book_name = $_POST['book_name'];
    $book_name = str_replace("'","''",$book_name);
    $image = $_POST['image'];
    $price = $_POST['price'];
    $user_id = $_SESSION['id'];
    $quantity = $_POST['quantity'];
    $sql = "SELECT * FROM cart WHERE c_email= '$user_id' AND book_id = $book_id";
    $result = mysqli_query($link,$sql);
    if(mysqli_num_rows($result)==1)
    {
      $status = '<div class="alert alert-danger" role="alert">Item is already added to cart</div>';
    }else {
```

```
    $sql = "INSERT INTO
cart(cart_id,book_id,book_name,img,price,total_price,quantity,c_email) VALUES(",$book_id,
    '$book_name','$image','$price','$price','$quantity','$user_id')";
    if(mysqli_query($link,$sql))
    {
     $status = '<div class="alert alert-success" role="alert">Item added to cart</div>';
    }else {
     $status = '<div class="alert alert-danger" role="alert">Cannot add item to cart</div>';
    }
   }
   echo $status;
   break;
 }
 ?>
```

4.4.1.4 Order review code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
$user_id = $_SESSION['id'];
$add = $_SESSION['address'];
//$checkout_id = $_SESSION['chkid'];
$output = '';
$sql = "SELECT * FROM pay_info where c_email = '$user_id' AND c_address = '$add'";
$result = mysqli_query($link,$sql);
if($result)
{
 while($row = mysqli_fetch_array($result))
 {
  $output .='<h5>Name:- '.$row['c_name'].'</h5>
            <br>
            <h5>Address:- '.$row['c_address'].'</h5>
            <br>
            <h5>Mobile Num:- '.$row['c_phone_no'].'</h5>';
 }
}
```

```
if (isset($_POST['pay'])) {
  $sql = "SELECT * FROM cart WHERE c_email = '$user_id'";
  $result = mysqli_query($link,$sql);
  if (!$result) {
    echo "Error occured";
  }else {
    $q = "SELECT * FROM pay_info WHERE c_address = '$add'";
    $res = mysqli_query($link,$q);
    while ($row = mysqli_fetch_array($res)) {
      $pay_id = $row['pay_id'];
      $address = $row['c_address'];
      $mobile = $row['c_phone_no'];
      $checkout = $row['checkout_id'];
    }
    $_SESSION['pay_id'] = $pay_id;
    $ord = rand().$pay_id;
  //  $i=0;
    while ($row = mysqli_fetch_array($result)) {
      $q1 = "INSERT INTO orders(sno, order_id, book_id, book_name, img, price,
quantity,total_price,
        c_email, date_of_purchase, status, payment_method, paid,c_address,c_phone_no)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
   //   $q2 = "INSERT INTO order_address(id, pay_id, order_id) VALUES(?,?,?)";
      $q3 = " DELETE FROM cart WHERE c_email = ?";
      $stmt1 = mysqli_prepare($link,$q1);
    //  $stmt2 = mysqli_prepare($link,$q2);
      $stmt3 = mysqli_prepare($link,$q3);
      mysqli_stmt_bind_param($stmt1,'isissssssssss',$param_sno,$param_orderid,$param_boo
kid,$param_bookname,$param_img,
      $param_price,$param_quantity,$param_total_price,$param_c_email,
      $param_dop,$param_status,$param_pm,$param_paid,$param_address,$param_mobile);
     // mysqli_stmt_bind_param($stmt2,'iis',$param_id,$param_pay_id,$param_orderid);
      mysqli_stmt_bind_param($stmt3,'s',$param_c_email);
      $param_sno = '';
      $param_orderid = $ord;
```

```
    $_SESSION['order_id']/*[$i]*/ = $param_orderid;
    $param_bookid = $row['book_id'];
    $param_bookname = $row['book_name'];
    $param_img = $row['img'];
    $param_price = $row['price'];
    $param_quantity = $row['quantity'];
    $param_total_price = $row['total_price'];
    $param_c_email = $user_id;
    $param_dop = date('d-m-Y');
    $param_status = "order placed";
    $param_pm = "Cash on delivery";
    $param_paid = 'no';
    $param_address = $address;
    $param_mobile = $mobile;
    $param_id = '';
    $param_pay_id = $pay_id;
    if(mysqli_stmt_execute($stmt1) && mysqli_stmt_execute($stmt3))
    {
      echo "<script>
          window.location.href = 'order_success.php';
      </script>";
    }
   //  $i++;
   }
  }
}
 ?>
```

4.4.1.5 Order success code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
$user_id = $_SESSION['id'];
$pay_id = $_SESSION['pay_id'];
$order_id = $_SESSION['order_id'];
```

```
if (!isset($_SESSION['order_id'])) {
  echo "<script>
      window.location.href = 'index.php';
  </script>";
}
$sql = "SELECT * FROM pay_info WHERE c_email ='$user_id' AND pay_id = $pay_id";
$result = mysqli_query($link,$sql);
while($row = mysqli_fetch_array($result))
{
  $address = $row['c_address'];
  $mobile = $row['c_phone_no'];

}
$total_price = 0;
$output =$out = '';
//foreach ($_SESSION['order_id'] as $key => $value) {
  $sql = "SELECT * FROM orders WHERE c_email='$user_id' AND order_id = '$order_id' ";
  $result = mysqli_query($link,$sql);
  while($row = mysqli_fetch_array($result))
  {
    //$order_id = $row['order_id'];
  //  $output = '';
    $output .='<tr><td>'.$row['book_name'].'</td>';
    $output .='<td> &#8377; '.$row['price'].'</td>';
    $output .='<td>'.$row['quantity'].'</td>';
    $output .='<td> &#8377; '.$row['total_price'].'</td></tr>';
    $total_price +=$row['total_price'];
    $dop = $row['date_of_purchase'];
    $status = $row['status'];
    $payment_method = $row['payment_method'];

    $out ='<div class="alert-secondary rounded-top p-2"><strong>ORDER ID :
'.$row['order_id'].'</strong></div>
        <table class="table table-dark">
        <tr><td>BOOK NAME</td>
```

```
        <td>PRICE</td>
        <td>QUANTITY</td>
        <td>TOTAL</td></tr>
   '.$output.'
        </table>
    ';
  }
//}
unset($_SESSION['order_id']);
 ?>
```

4.4.2 Admin:

4.4.2.1 Login code:

<?php

```
$link = mysqli_connect("localhost","root","","bookstore");
$username = $password = "";
$username_err=$password_err =$error = "";
if (isset($_POST['submit'])) {
if (empty($_POST['username'])) {
  $username_err = "please enter username";
}else{
  $username = test_input($_POST['username']);
}
if (empty($_POST['password'])) {
  $password_err = "please enter password";
}else{
  $password = test_input($_POST['password']);
}
if (empty($username_err) && empty($password_err)) {
  $sql = "SELECT * FROM admin WHERE email = '$username' AND password =
'$password'";
  $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0) {
    $_SESSION['aloggedin'] = true;
    header('location:profile.php');
  }else {
```

```
  $error = "invalid login details";
 }
}
}
function test_input($data){
 $data = trim($data);
 $data = stripslashes($data);
 $data = htmlspecialchars($data);
 return $data;
}
 ?>
```

4.4.2.2 View registered users code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
$output ="";

 if (isset($_POST['q'])) {
  $username = test_input($_POST['username']);
  $sql= "SELECT * FROM registration WHERE c_email='$username'";
 $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0)
  {
    while ($row = mysqli_fetch_array($result)) {
      $output = '<table class="table">
        <tr>
          <th>Username</th>
          <th>Name</th>
          <th>Mobile</th>
        </tr>
        <tr>
```

```
          <td>'.$row['c_email'].'</td>
          <td>'.$row['c_name'].'</td>
          <td>'.$row['c_phone_no'].'</td>
        </tr>
      </table>';
    }
  }else {
    $output = '<div class="alert alert-danger">No record found</div>';
  }
}
function test_input($data){
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
 ?>
```

4.4.2.3 Add book code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
 $bookname_error=$author_error=$price_error=$category_error=$image_error="";
 $bookname = $author = $price = $category = $image= $status = "";
 if (isset($_POST['submit'])) {
  if (empty($_POST['book_name'])) {
    $bookname_error = "please enter book name";
  }else {
    $bookname= test_input($_POST['book_name']);
```

```
  $name_pattern='/^[a-zA-Z ]+$/';
  if (!preg_match($name_pattern,$bookname)) {
    $bookname_error ="please enter valid book name";
  }
}
  //validate author
  if (empty($_POST['author'])) {
    $author_error = "please enter author name";
  }else {
    $author= test_input($_POST['author']);
    $author_pattern='/^[a-zA-Z ]+$/';
    if (!preg_match($author_pattern,$author)) {
      $author_error ="please enter valid author name";
    }
  }
  //validate price
  if (empty($_POST['price'])) {
    $price_error = "please enter price";
  }else {
    $price= test_input($_POST['price']);
    $price_pattern='/^[0-9]+$/';
    if (!preg_match($price_pattern,$price)) {
      $price_error ="please enter valid price";
    }
  }
  if (empty($_POST['category'])) {
```

```
    $category_error = "please enter category";
   }else {
    $category= test_input($_POST['category']);
    $category_pattern='/^[a-zA-Z ]+$/';
    if (!preg_match($category_pattern,$category)) {
     $category_error ="please enter valid category";
    }
   }
     if (!isset($_FILES['book_img'])) {
    $image_error = "please select a image";
   }
   else {
    $target = "images/";
    $file_name = $_FILES['book_img']['name'];
    $file_type = $_FILES['book_img']['type'];
    $file_size = $_FILES['book_img']['size'];
    $tmp_name = $_FILES['book_img']['tmp_name'];
    $allowed = array('jpg' =>'image/jpg' ,'jpeg'=>'image/jpeg' );
if (!in_array($file_type,$allowed)) {
     $image_error = "please select jpg/jpeg file";
    }
    $maxsize = 1*1024*1024;
    if ($file_size>$maxsize) {
     $image_error="file size is out of range";
    }
        if (in_array($file_type,$allowed) && $file_size<$maxsize &&
      $_FILES['book_img']['error']===0) {
```

```
            $newname = rand().$file_name;
          $target = $target.$newname;
          $image = $target;
         move_uploaded_file($tmp_name,$target);
  }else {
        $image_error = "some error occured";
       }
     }
 if (empty($bookname_error) && empty($author_error) && empty($price_error) &&
          empty($category_error) && empty($image_error)) {
       $sql = "INSERT INTO book values('','$bookname','$image','$author','$price','$category')";
         if (mysqli_query($link,$sql)) {
        $status = '<div class="alert alert-success">Book added successfully</div>';
       }else {
         $status = '<div class="alert alert-danger">Error occured</div>';
       }
     }
}
 function test_input($data){
   $data = trim($data);
   $data = stripslashes($data);
   $data = htmlspecialchars($data);
   return $data;
}
  ?>
```

4.4.2.4 View book code:

```
<?php
```

```
$link = mysqli_connect("localhost","root","","bookstore");
 $output ="";
 if (isset($_POST['updatebook'])) {
  $_SESSION['bookid'] = $_POST['update_book'];
  echo "<script>
     window.location.href = 'update_books.php';
  </script>";
 }
 if (isset($_POST['q'])) {
  $book_name = test_input($_POST['book_name']);
  $sql= "SELECT * FROM book WHERE book_name LIKE '%$book_name%'";
    $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0)
  {
   $output .= '<table class="table">
     <tr>
      <th>Book_id</th>
      <th>Bookname</th>
      <th>Author</th>
      <th>Price</th>
      <th>Image</th>
     </tr>';
   while ($row = mysqli_fetch_array($result)) {
$output .= '
      <tr>
       <td>'.$row['book_id'].'</td>
```

```
        <td>'.$row['book_name'].'</td>
        <td>'.$row['author'].'</td>
        <td>'.$row['price'].'</td>
        <td><img src="'.$row["img"].'" height="100" width="100">   
        <form method="post">
        <input type="hidden" name="update_book" value="'.$row['book_id'].'">
            <button type ="submit" class="btn btn-sm btn-outline-primary"
        name="updatebook">Update</button>
        </form>
        </td>
       </tr>';
 }
   $output .='</table>';
 }else {
   $output = '<div class="alert alert-danger">No record found</div>';
  }
}
if (isset($_POST['all'])) {
  $sql="SELECT * FROM book";
  $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0)
  {
   $output .= '<table class="table">
     <tr>
     <th>Book_id</th>
     <th>Bookname</th>
     <th>Author</th>
```

```
    <th>Price</th>
    <th>Image</th>
    <th></th>
    </tr>';
  while ($row = mysqli_fetch_array($result)) {
        $output .= '
    <tr>
    <td>'.$row['book_id'].'</td>
    <td>'.$row['book_name'].'</td>
    <td>'.$row['author'].'</td>
    <td>'.$row['price'].'</td>
    <td><img src="'.$row["img"].'" height="100" width="100">   </td>
    <td>
    <form method="post">
    <input type="hidden" name="update_book" value="'.$row['book_id'].'">
        <button type ="submit" class="btn btn-sm btn-outline-primary"
      name="updatebook">Update</button>
    </form>
    </td>
    </tr>';
  }
  $output .='</table>';
 }else {
  $output = '<div class="alert alert-danger">No record found</div>';
 }
}
function test_input($data){
```

```
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
 }
 ?>
```

4.4.2.5 Delete book code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
 $output="";
 if (isset($_POST['q'])) {
  $book_name = test_input($_POST['book_name']);
  $sql= "SELECT * FROM book WHERE book_name = '$book_name'";
    $result = mysqli_query($link,$sql);
  if (mysqli_num_rows($result)>0)
  {
    $sql1="DELETE FROM book WHERE book_name='$book_name'";
    if (mysqli_query($link,$sql1)) {
      $output = '<div class="alert alert-success">Book deleted successfully</div>';
    }
  }else {
    $output = '<div class="alert alert-danger">No record found</div>';
  }
 }
  function test_input($data){
    $data = trim($data);
    $data = stripslashes($data);
```

```
    $data = htmlspecialchars($data);
    return $data;
  }
?>
```

4.4.2.6 Oder history code:

```
<?php
$link = mysqli_connect("localhost","root","","bookstore");
  if (isset($_POST['q'])) {
  $username = test_input($_POST['username']);
   $sql = "SELECT DISTINCT order_id FROM orders WHERE c_email = '$username' ORDER
        BY sno DESC" ;
 $result = mysqli_query($link,$sql);
 $total_orders = mysqli_num_rows($result);
  while($row = mysqli_fetch_array($result)){
   $order_id = $row['order_id'];
                $out1 = '';
          $ql = "SELECT * FROM orders WHERE c_email = '$username' AND order_id =
'$order_id'";
                $res = mysqli_query($link,$ql);
                while($row = mysqli_fetch_array($res))  {
                  $address = $row['c_address'];
                  $mobile = $row['c_phone_no'];
                  $dob = $row['date_of_purchase'];
                  $status = $row['status'];
                 $out1 .= '<tr>
                 <td>'.$row['book_name'].'</td>
                 <td> &#8377; '.$row['price'].'</td>
```

```
                <td>'.$row['quantity'].'</td>
                <td> &#8377; '.$row['total_price'].'</td>
                </tr>';
                 }
                $out = '<div class="alert-secondary rounded-top p-2">
                    <form method="post">
                    <strong>Username : '.$username.'</strong><br>
                    <strong>Delivery Address : '.$address.'</strong><br>
                    <strong>Mobile : '.$mobile.'</strong><br>
                    <strong>Date Of Purchase  : '.$dob.'</strong><br>
                    <strong>Status : '.$status.'</strong><br>
                    <strong>Order Id : '.$order_id.'</strong></form>
                    </div>
                    <table class="table table-dark">
                    <tr><th class="w-25">BOOK NAME</th>
                    <th>PRICE</th>
                    <th>QUANTITY</th>
                    <th>TOTAL</th></tr>
                    '.$out1.'
                    </table><br>';
echo $out;
            }
           }
           if (isset($_POST['all'])) {
            $sql = "SELECT DISTINCT order_id,c_email FROM orders ORDER BY sno DESC"
        ;
            $result = mysqli_query($link,$sql);
```

```
$total_orders = mysqli_num_rows($result);
while($row = mysqli_fetch_array($result)){
    $order_id = $row['order_id'];
    $username = $row['c_email'];
    $out1 = '';
    $ql = "SELECT * FROM orders WHERE  order_id = '$order_id'";
    $res = mysqli_query($link,$ql);
    while($row = mysqli_fetch_array($res))   {
     $address = $row['c_address'];
     $mobile = $row['c_phone_no'];
     $dob = $row['date_of_purchase'];
     $status = $row['status'];
    $out1 .= '<tr>
    <td>'.$row['book_name'].'</td>
    <td> &#8377; '.$row['price'].'</td>
    <td>'.$row['quantity'].'</td>
    <td> &#8377; '.$row['total_price'].'</td>
    </tr>';
     }
    $out = '<div class="alert-secondary rounded-top p-2">
        <form method="post">
        <strong>Username : '.$username.'</strong><br>
        <strong>Delivery Address : '.$address.'</strong><br>
        <strong>Mobile : '.$mobile.'</strong><br>
        <strong>Date Of Purchase  : '.$dob.'</strong><br>
        <strong>Status : '.$status.'</strong><br>
```

```
                    <strong>Order Id : '.$order_id.'</strong>
                    </form>
                    </div>
                    <table class="table table-dark">
                    <tr><th class="w-25">BOOK NAME</th>
                    <th>PRICE</th>
                    <th>QUANTITY</th>
                    <th>TOTAL</th></tr>
                    '.$out1.'
                    </table>';
echo $out;
            }
           }
```

4.5 Discussion of Results

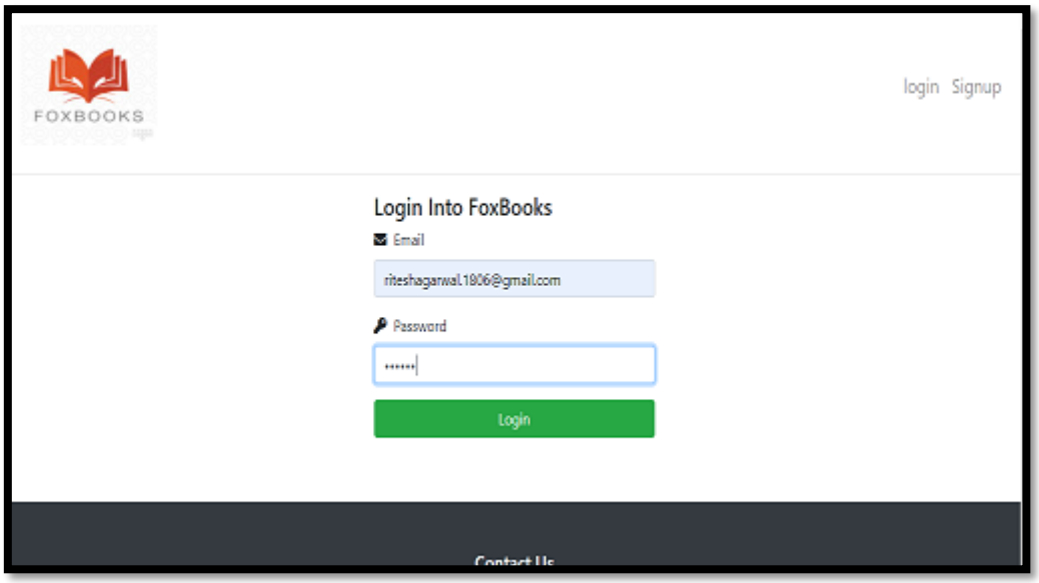

Figure 4.2 Homepage with login prompt

The above Figure 4.2 is the snapshot of the homepage with the login prompt. If the customer has already registered with the application, he can login using his email ID which he used to register along with the valid password also which he set during sign up.

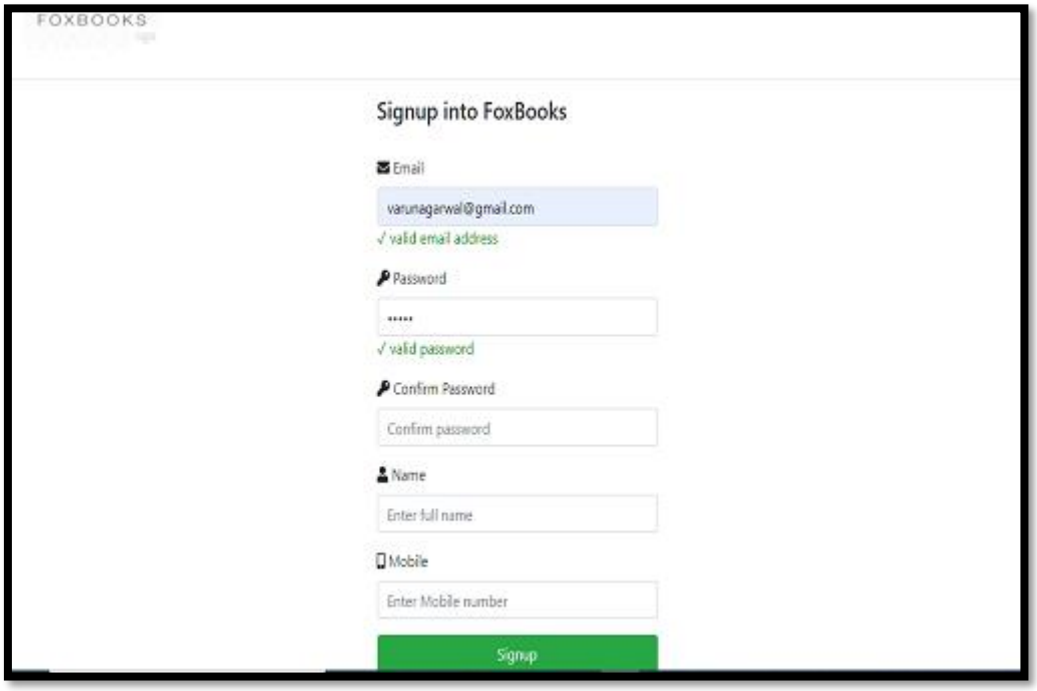

Figure 4.3 Signup Prompt

The above Figure 4.3 is the snapshot of the signup prompt, if a new user wants to register with the website, he can use this form to register. It accepts the email ID, password, and his contact details like name, phone number.

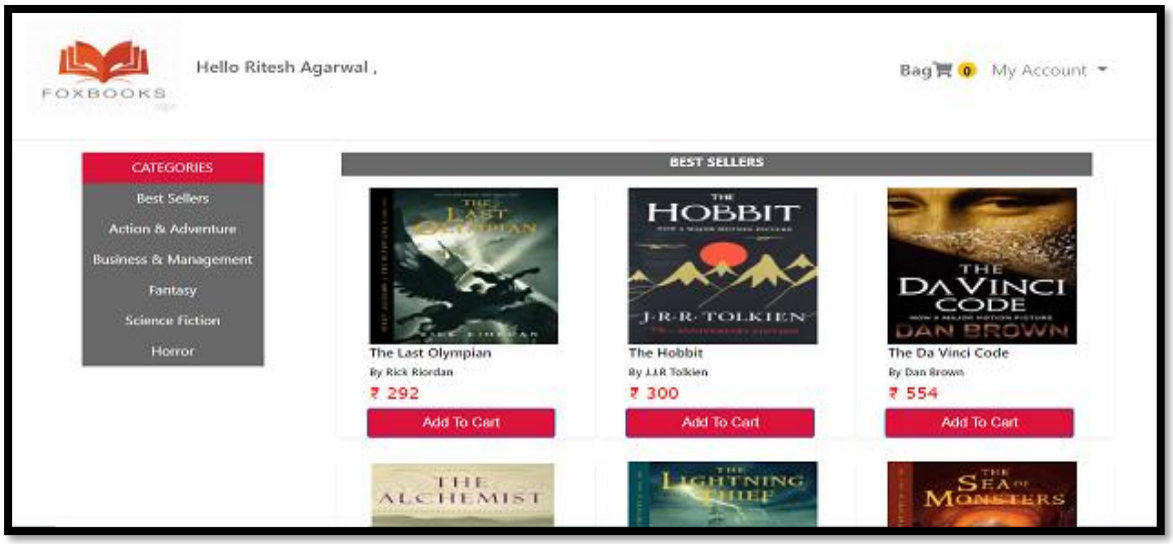

Figure 4.4 index page of the bookstore

The above figure 4.4 shows the index page of the bookstore from where customers can select the books of the desired category he wants to purchase. To add books to the cart the customer has to log in first and if he is already logged in then he can add books by clicking

on the add to cart button.

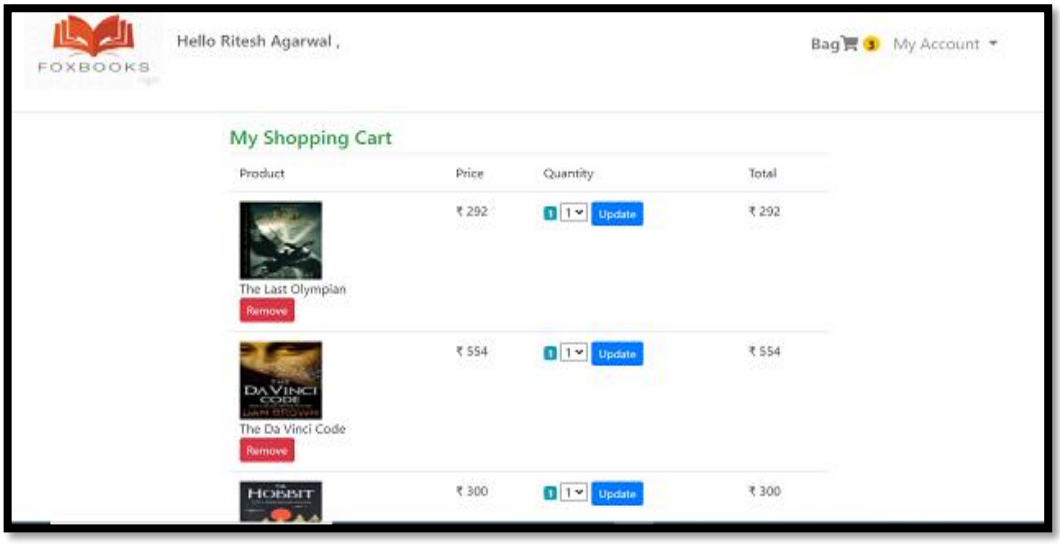

Figure 4.5 Shopping cart

This page shown in the above Figure 4.5 displays the list of books which the customer wishes to buy. In the cart he can remove the book if he/she didn't want it by clicking remove button. The customer can also buy multiple copies of a single book by selecting the number and clicking update.

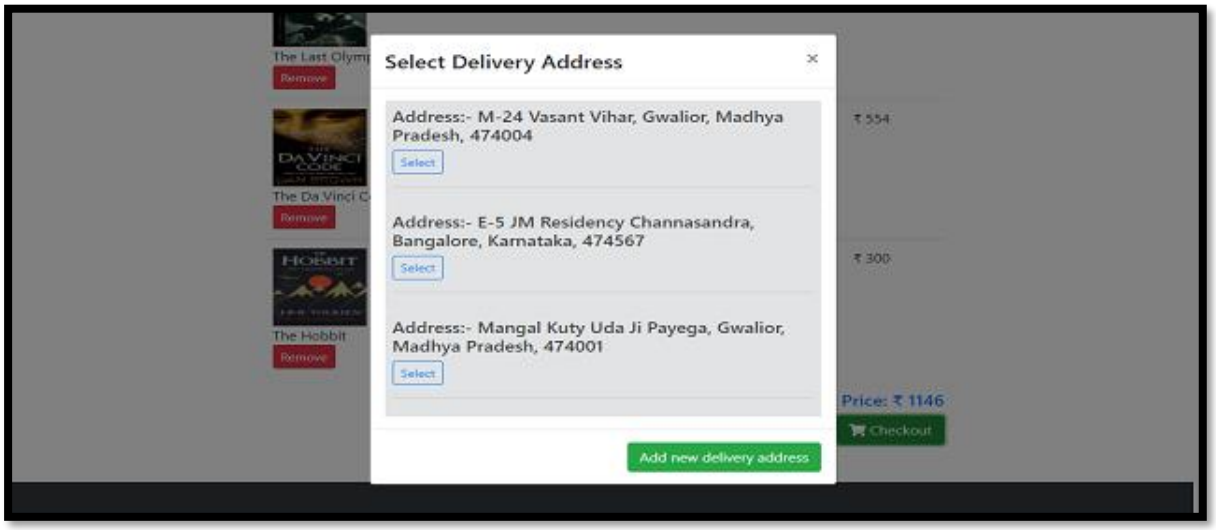

Figure 4.6 prompt to select/add delivery address

The above Figure 4.6 show a modal that prompts the user to select the delivery address on which he wants the order to be delivered. This modal appears when the customer clicks the checkout button on the cart page. This modal also provides the functionality if the customer wants to add a new delivery address.

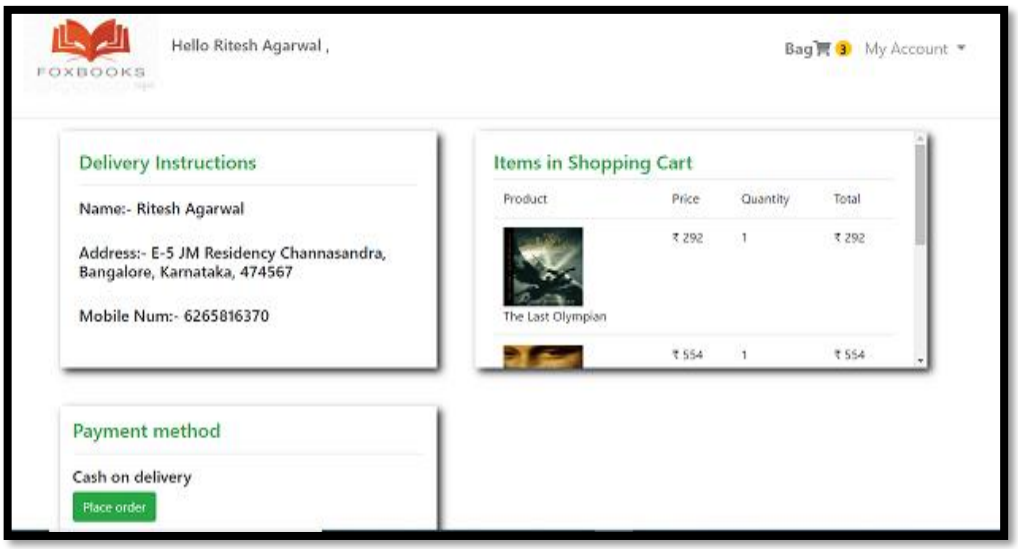

Figure 4.7 Order review page

The above Figure 4.7 shows the order review page where the customer can review his delivery details and the order before placing the order.

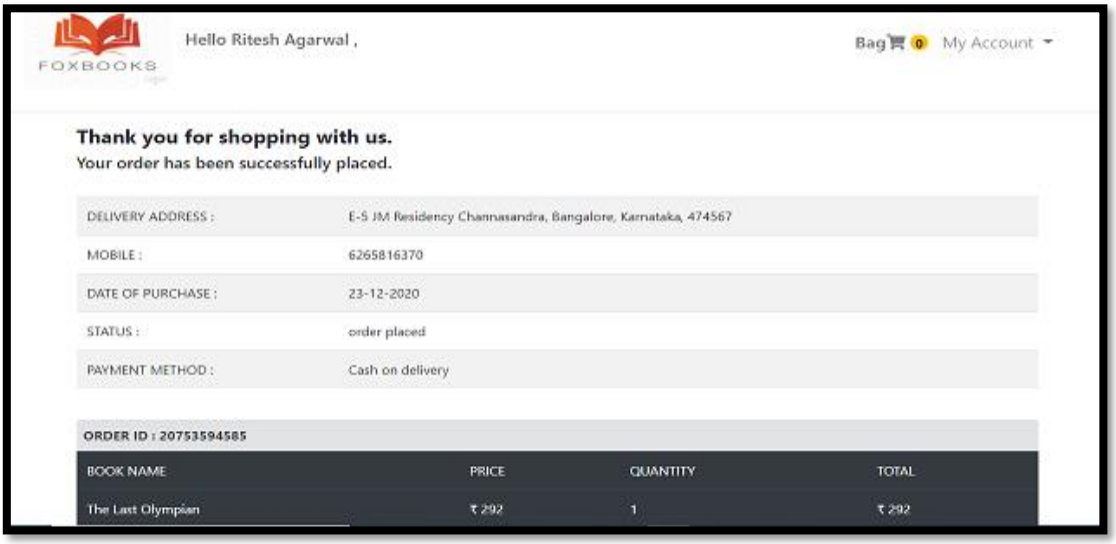

Figure 4.8 Order success page

After clicking the place order button the customer will be directed to the order success page which is shown in the Figure 4.8. It shows the date of purchase of the order and status that the order has been successfully placed.

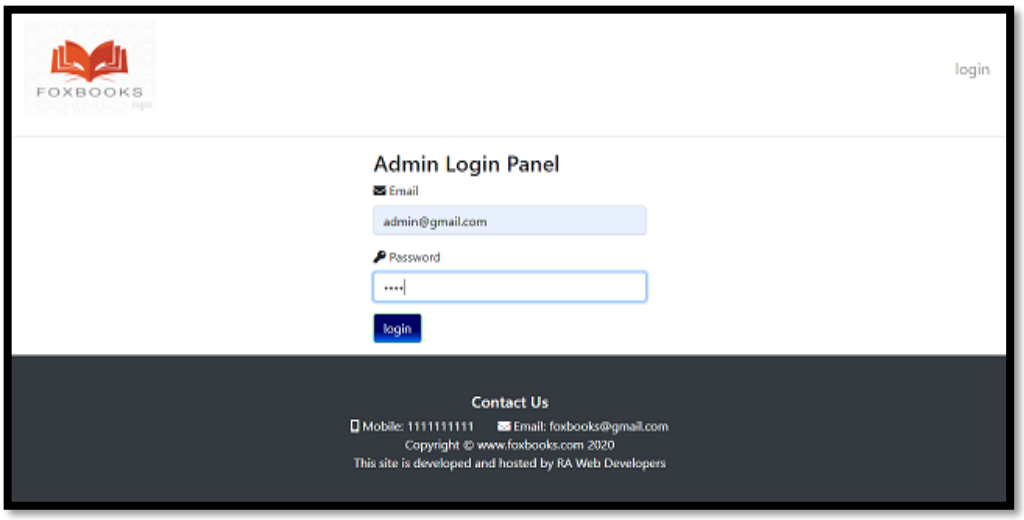

Figure 4.9 prompt for admin login

The above Figure 4.9 is the snapshot of the admin login prompt. If the admin has registered with the application, he can login using his email ID along with the valid password.

Figure 4.10 Admin login panel

The above Figure 4.10 is the snapshot of the admin login panel in which the admin has different functionalities. The admin can view the users who are registered with the application. He/she can view the books that are available in the bookstore along with that he can add new books, delete and update the available books. The admin can also view the orders that are successfully placed by the customers.

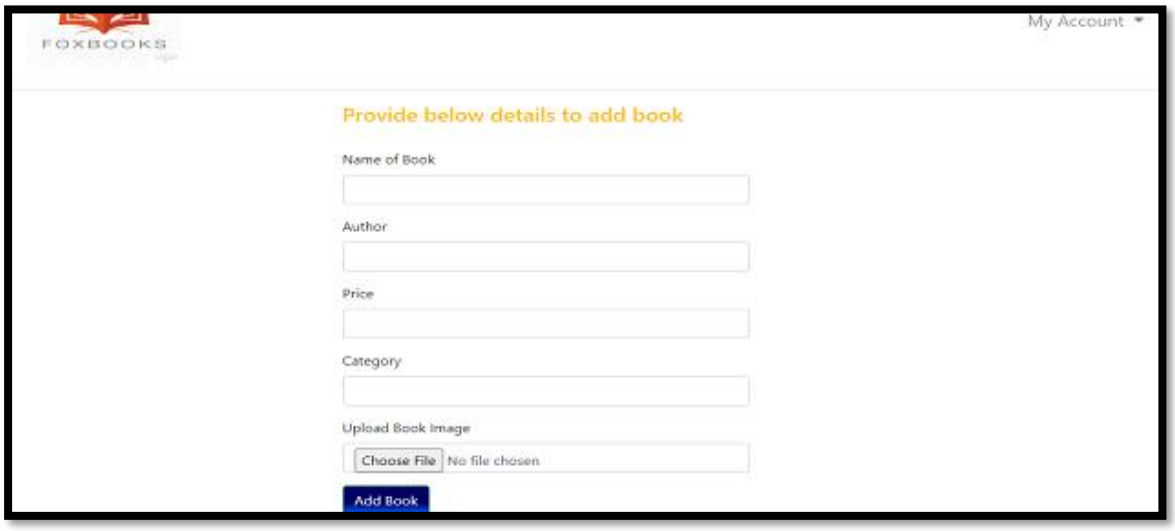

Figure 4.11 Add Book page

The above Figure 4.11 is the add book page in which the admin can add new books in the application by specifying the name of book, author, price, category, image.

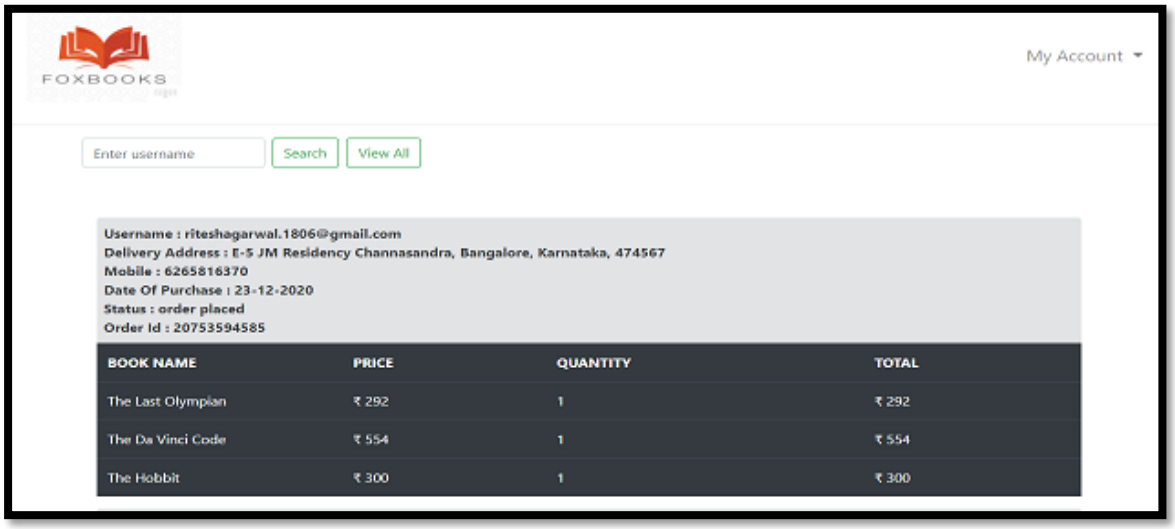

Figure 4.12 Order History page

The above Figure 4.12 is the snapshot of the order history page where the admin can view the orders placed by a particular user by searching the customer's email ID. He can also view the orders placed by all the registered customers by clicking on the view all button

4.6 Application of project

These types of Online Bookstore Management Systems are examples of flourish technology has had on the world. These help customers to purchase books from their laptops, mobiles phones or tablets. They are easy to use and get the job done in very few steps. The old methods of standing and searching for hours is gone.

Major applications of Online Bookstore Management Systems are:

1. Opting for an online business instead of physical store can save a lot of time and hard work of searching for an ideal store at an ideal location.
2. Online bookstore is much easier to operate as compared to a physical one.
3. Online bookstore has larger reach as compared to physical store.
4. Online bookstore creates more job opportunities than physical store.

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

This project was an attempt to make the structure and working of an Online Bookstore Management System simpler and user-friendly. This was an attempt to make it similar to the real world implementation. In this scenario, all the undertakings of the Online Bookstore Management System were achieved in a constructive manner. The application is tested to ensure its effectiveness and provide error free functionality to the end user. Given the right guidance and support its applications and availability can be enhanced.

5.2 Future Enhancements

1. Hosting it on an online web server.
2. Email and message notifications could be sent to the customers on successful purchase of books.
3. Introduction of new payment methods like paying through credit/debit cards, using UPI for transaction etc.
4. Providing a way for the customers to cancel and track their orders.
5. To have a page where customer can give their feedback so that the admin can look into these and take actions accordingly.
6. Providing a functionality where customers can sell their used books.

REFERENCES

[1] https://www.google.com

[2] https://www.w3schools.com

[3] www.stackoverflow.com

[4] www.mysql.com/doc

[5] https://docs.microsoft.com

[6] https://www.geeksforgeeks.org