VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAUM-590014

A Mini-Project Report

On

"ONLINE SHOPPING DATABASE MANAGEMENT SYSTEM"

A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belgaum.

Submitted By:

UPENDRA GUPTA (1AM18CS186)

Under the Guidance of

Mr. MUSHTAQ AHMED D. M

(Asst.Prof.Dept.of CSE)

Department of Computer Science and Engineering

AMC Engineering College,

18th K.M, Bannerghatta Main Road, Bangalore - 560 083

2020-2021

AMC Engineering College,

18th K.M, Bannerghatta Main Road, Bangalore-560 083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini-project work entitled **"ONLINE SHOPPING DATABASE MANAGEMENT SYSTEM"** has been successfully carried out by **UPENDRA GUPTA (1AM18CS186)** bona fide students of **AMC Engineering College** in partial fulfilment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Guide:

Mr. MUSHTAQ AHMED D. M	**Dr. LATHA C A**	**Dr. A.G NATARAJ**
Asst. Prof. Dept. of CSE	**HOD, Dept. of CSE**	**Principal, AMCEC**

Examiners: **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **AMC ENGINEERING COLLEGE** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express sincere thanks to our respected chairman **Dr. K. R. Paramahamsa** and beloved principal **Dr. A.G. Nataraj** for all their support.

We express our deepest gratitude and special thanks to **Dr. Latha C A**, **H.O.D, Dept. Of Computer Science Engineering**, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project guide, **Mr. MUSHTAQ AHMED D. M, Assistant Prof. Dept. Of Computer Science Engineering.**

NAME USN

UPENDRA GUPTA (1AM18CS186)

ABSTRACT

The entitled project "**ONLINE SHOPPING DATABASE MANAGEMENT SYSTEM**" is a Web based shopping system for an existing shop.

The project objective is to deliver the online shopping application to public. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as Flipkart or Amazon. Since the application is available on the internet it is easily accessible and always available.

CONTENTS

Title **Page No**

Chapter 1

INTRODUCTION

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform.

Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop.

Online shopping management system describes the complete process of purchasing various products by a customer from the online portal. “ONLINE SHOPPING DATABASE MANAGEMENT SYSTEM” has been designed to computerize the following functions that are performed by the system:

- Product details
- Payment details
- Order summary
- Add and display new products for customers
- Customer details

1.1 OBJECTIVES

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

1.2 PROJECT OVER VIEW

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store).

The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which is deployed at the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security.

1.3 SCOPE

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy.

If shops are providing an online portal where their customers can enjoy easy shopping From anywhere, the shops won't be losing any more customers to the trending online Shops such as Flipkart or Amazon. Since the application is available in the Smartphone it is easily accessible and always available

.

Chapter 2

SYSTEM SPECIFICATION

2.1 HARDWARE REQUIREMENTS

- Server side Hardware
- Hardware Recommended by the all the software needed
- Communication hardware to server client request.
- Client side Hardware
- Hardware recommended by respective client operating system & Web browser.

2.2 SOFTWARE REQUIREMENTS

- OS: Microsoft Windows 10, 64bit
- Server Side Software
- Web Server Software
- Server Side Scripting Tools
- Database Tools

CODING LANGUAGES USED:

- **FRONT END: HTML, CSS, JAVASCRIPT**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With **Cascading Style Sheets** (CSS) and **JavaScript** it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- **BACK END: PHP AND PHPMYADMIN**

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. An application software called Navicat was used to design the tables in MySQL.

Chapter 3

DESIGN

3.1 Description of Online Shopping Management Database System

- The details of orders, customer, and products are stored into the respective tables.
- Each entity (Customer, Product, Category, Order, and Payment) contains primary key.
- There is one-to-one and one-to-many relationships available between Payment, Order, Customers, and Products.
- All the entities are normalized and reduce duplicity of records.
- Indexing is implemented on each tables of Online Shopping Management System tables for fast query execution.

There are many brand product branches in a country for a single owner.

Each customer's name, address, contacts are stored. Each customer is provided with a unique customer code.

The customer logins to his account and books a product according to his choice. The customer is needed to give details of his email and card details for booking the product. After booking an item the customer can view this details with the number or email address provided during the payment and carry out the process of delivering the products safely to the address given by the customer.

The admin is responsible for keeping track of the product which are added or deleted by admin and order details for the item purchased.

STEP 1:

Entities are:

1. Customer
2. Product
3. Category
4. Payment

5. Order
6. Cart
7. Brands

STEP 2:

RELATIONS:

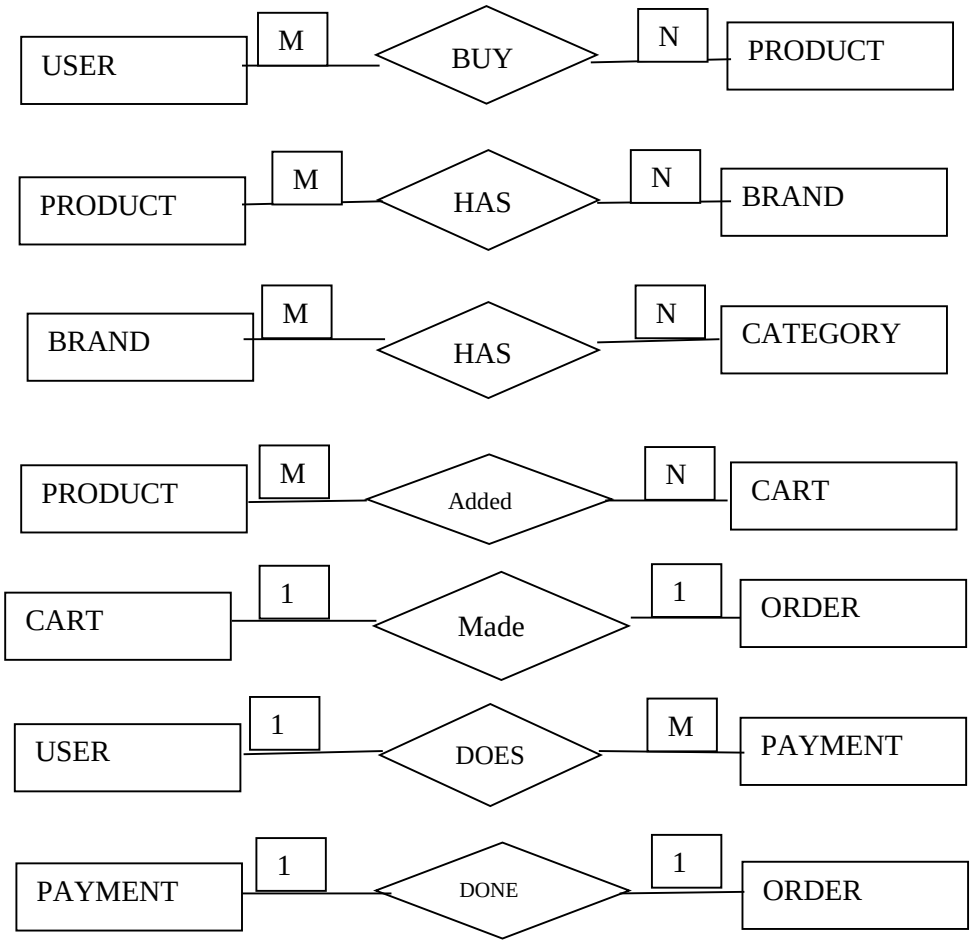

Figure 3.1 : Relation of entities

STEP 3:

Key attributes:

USER- user_id

PRODUCT-product_id

PAYMENT- trx_id

ORDER- order_id

CART- id

CATEGORY- cat_id

BRAND- brand_id

STEP 4:

Other attributes:

User- f_name, l_name, email, password, mobile, address

Order- order_id, user_id, product_id, qty, trx_id, p_status

Admin- admin_id, admin_name, admin_password, admin_email

Cart- id, p_id, ip_add, user_id, qty

Order_info- order_id, user_id, f_name, email, address, city, state, zip, cardname, cardnumber, expdate, prod_count, total_amt, cvv, redeem_points

SCHEMA DIAGRAM

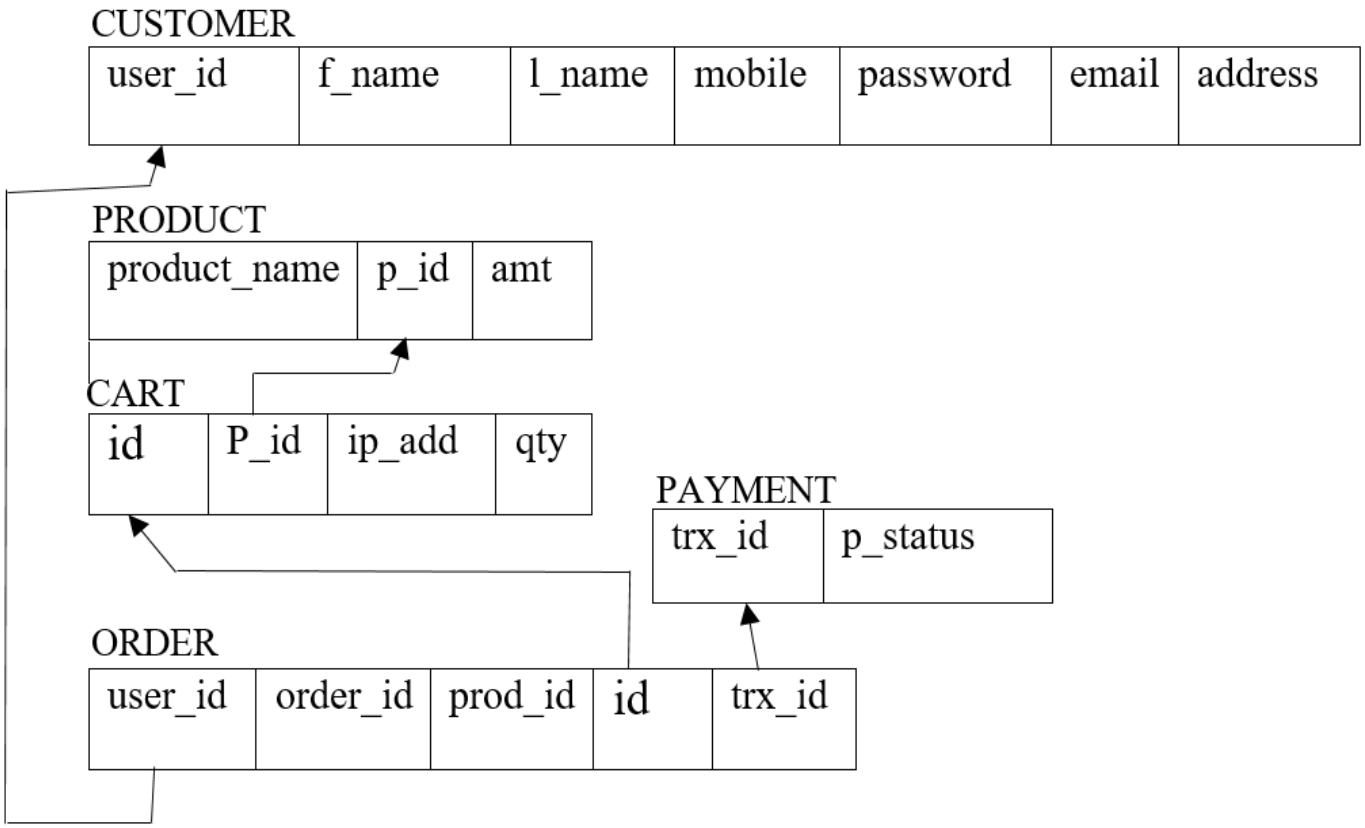

Figure 3.2 : Relation schema for online shopping

STEP 5:

ER DIAGRAM

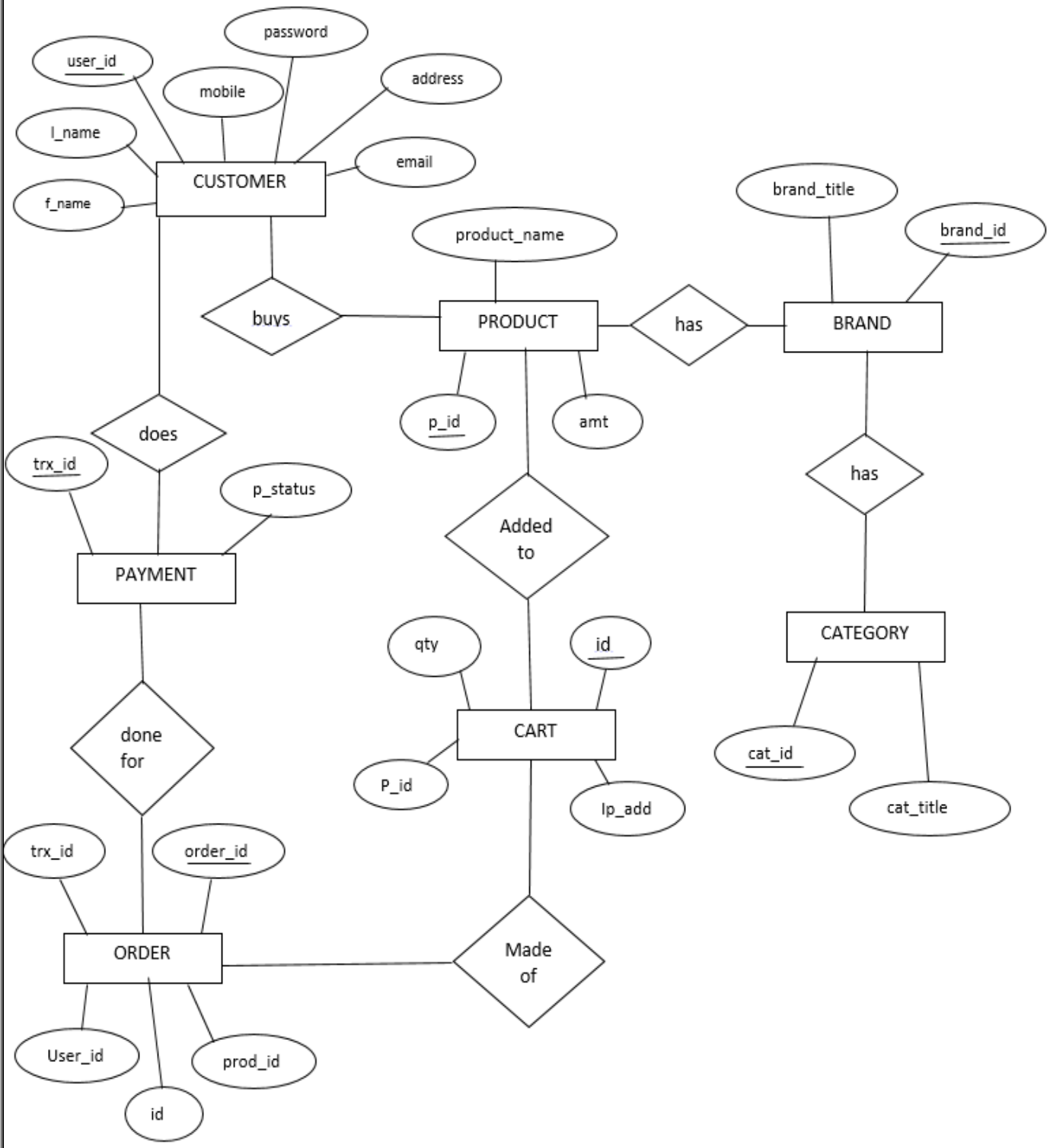

Figure 3.3 : ER Diagram

3.2 NORMALISATION

The basic Objectives of normalization are to reduce redundancy, which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of data stored. Relations are normalized so that when relation in the database are to be altered during the lifetime of the database, information is not lost or introduces inconsistencies. The type of alterations normally needed for relation is:

Insertion of new data values to relation. This should be possible without being forced to leave blank fields for some attributes.

Deletion of a tuple, namely, a row of a relation. This should be possible without losing vital information unknowingly.

Functional Dependency:

As the concept of dependency is very important, it is essential that it should be understood first and then proceed to the idea of normalization. There is no fool-proof algorithmic method of identifying dependency.

Properties of normalized relations:

Ideals relation after normalization should have the following properties:

No data values should be duplicated in different rows unnecessarily.

A value must be specified (and required) for every attribute in a row.

Each relation should be self-contained. In other words, if a row from a relation is deleted, important information should not be accidentally lost.

When a row is added to a relation, other relations in the database should not be affected.

A value of an attribute in a tuple may be changed independent of other tuples in the relation and other relations.

Consider the PAYMENT table (refer to the schema diagram on page 13).

The prime attributes identified are the attributes which is part of candidate key.

The non-prime attributes are not part of primary key. There is only one primary key in each table and all the non-key attributes can be derived from the primary key. There is no partial dependency and hence the table is in 2NF.

TRIGGERS

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.

BEFORE and AFTER :

BEFORE triggers run the trigger action before the triggering statement is run. AFTER triggers run the trigger action after the triggering statement is run.

```
CREATE TRIGGER `redeem_points`
BEFORE INSERT ON `orders_info`
FOR EACH ROW
SET new.redeem_points = (new.total_amt*0.01);
```

PROCEDURES:

Stored Procedures are created to perform one or more DML operations on Database. It is nothing but the group of SQL statements that accepts some input in the form of parameters and performs some task and may or may not return a value .

The most important part is parameters. Parameters are used to pass values to the Procedure. There are 3 different types of parameters, they are as follows:

IN:

This is default parameter of the procedure . It always receives the values from calling program.

OUT:

This parameter always sends the values to the calling program.

IN OUT :

This parameter performs both the operations.It receives values from as well as sends the values to the calling program.

```
DELIMITER $$

---Procedures

CREATE DEFINER=`root`@`localhost` PROCEDURE `getcat` (IN `cid` INT)  SELECT * FROM categories WHERE cat_id=cid$$

DELIMITER ;
```

Chapter 4

IMPLEMENTATION AND CODING

4.1 SOURCE CODE

Store.php

```
<?php
include 'header.php';
?>
    <script id="jsbin-javascript">
(function (global) {
   if(typeof (global) === "undefined")
   {
      throw new Error("window is undefined");
   }
   var _hash = "!";
   var noBackPlease = function () {
      global.location.href += "#";
      // making sure we have the fruit available for juice....
      // 50 milliseconds for just once do not cost much (^__^)
      global.setTimeout(function () {
         global.location.href += "!";
      }, 50);
   };
   // Earlier we had setInerval here....
   global.onhashchange = function () {
      if (global.location.hash !== _hash) {
         global.location.hash = _hash;
      }
   };
   global.onload = function () {
      noBackPlease();
      // disables backspace on page except on input fields and textarea..
      document.body.onkeydown = function (e) {
         var elm = e.target.nodeName.toLowerCase();
         if (e.which === 8 && (elm !== 'input' && elm  !== 'textarea')) {
            e.preventDefault();
         }
```

```
 // stopping event bubbling up the DOM tree..
          e.stopPropagation();
       };
    };
})(window);
</script>
     <div class="main main-raised">

       <div class="section">
          <!-- container -->
          <div class="container">
             <!-- row -->
             <div class="row">
                <!-- ASIDE -->
                <div id="aside" class="col-md-3">
                   <!-- aside Widget -->
                   <div id="get_category">
                   </div>
                   <!-- /aside Widget -->

                   <!-- aside Widget -->
                   <div class="aside">
                      <h3 class="aside-
title"style="background:#333333; color:whitesmoke; padding:10px; border-radius:5px;">Price</h3>
                         <div class="price-filter">
                            <div id="price-slider" class="noUi-target noUi-ltr noUi-
horizontal"><div class="noUi-base"><div class="noUi-origin" style="left: 0%;"><div class="noUi-
handle noUi-handle-lower" data-handle="0" tabindex="0" role="slider" aria-
orientation="horizontal" aria-valuemin="0.0" aria-valuemax="100.0" aria-valuenow="0.0" aria-
valuetext="1.00" style="z-index: 5;"></div></div><div class="noUi-
connect" style="left: 0%; right: 0%;"></div><div class="noUi-
origin" style="left: 100%;"><div class="noUi-handle noUi-handle-upper" data-
handle="1" tabindex="0" role="slider" aria-orientation="horizontal" aria-valuemin="0.0" aria-
valuemax="100.0" aria-valuenow="100.0" aria-valuetext="999.00" style="z-
index: 4; color:#333333;"></div></div></div></div>
                         <div class="input-number price-min">
                            <input id="price-min" type="number">
                            <span class="qty-up">+</span>
                            <span class="qty-down">-</span>
                         </div>
                         <span>-</span>
                         <div class="input-number price-max">
                            <input id="price-max" type="number">
                            <span class="qty-up">+</span>
                            <span class="qty-down">-</span>
                         </div>
                      </div>
                   </div>
                   <!-- /aside Widget -->

                   <!-- aside Widget -->
```

```
        <div id="get_brand">
                </div>
                <!-- /aside Widget -->

                <!-- aside Widget -->
                <div class="aside">
                   <h3 class="aside-
title" style="background:#333333; color:whitesmoke; padding:10px; border-
radius:5px;">Top selling</h3>
                   <div id="get_product_home">
                      <!-- product widget -->

                      <!-- product widget -->
                   </div>
                </div>
                <!-- /aside Widget -->
             </div>
             <!-- /ASIDE -->

             <!-- STORE -->
             <div id="store" class="col-md-9">
                <!-- store top filter -->
                <div class="store-filter clearfix">
                   <div class="store-sort">
                      <label>
                         Sort By:
                         <select class="input-select">
                            <option value="0">Popular</option>
                            <option value="1">Position</option>
                         </select>
                      </label>

                      <label>
                         Show:
                         <select class="input-select">
                            <option value="0">20</option>
                            <option value="1">50</option>
                         </select>
                      </label>
                   </div>
                   <ul class="store-grid">
                      <li class="active"><i class="fa fa-th"></i></li>
                      <li><a href="#"><i class="fa fa-th-list"></i></a></li>
                   </ul>
                </div>
                <!-- /store top filter -->

                <!-- store products -->
                <div class="row" id="product-row">
                <div class="col-md-12 col-xs-12" id="product_msg">
             </div>
```

```
                    <!-- product -->
                        <div id="get_product">
                        <!--Here we get product jquery Ajax Request-->
                    </div>

                        <!-- /product -->
                    </div>
                    <!-- /store products -->

                    <!-- store bottom filter -->
                    <div class="store-filter clearfix">
                        <span class="store-qty">Showing 20-100 products</span>
                        <ul class="store-pagination" id="pageno">
                            <li ><a class="active" href="#aside">1</a></li>

                            <li><a href="#"><i class="fa fa-angle-right"></i></a></li>
                        </ul>
                    </div>
                    <!-- /store bottom filter -->
                </div>
                <!-- /STORE -->
            </div>
            <!-- /row -->
        </div>
        <!-- /container -->
    </div>
</div>
<?php
include "footer.php";
?>
```

Checkout.php

```
<?php
include "db.php";

include "header.php";


?>

<style>

.row-checkout {
  display: -ms-flexbox; /* IE10 */
  display: flex;
  -ms-flex-wrap: wrap; /* IE10 */
  flex-wrap: wrap;
  margin: 0 -16px;
}
```

```
.col-25 {
  -ms-flex: 25%; /* IE10 */
  flex: 25%;
}

.col-50 {
  -ms-flex: 50%; /* IE10 */
  flex: 50%;
}

.col-75 {
  -ms-flex: 75%; /* IE10 */
  flex: 75%;
}

.col-25,
.col-50,
.col-75 {
  padding: 0 16px;
}

.container-checkout {
  background-color: #f2f2f2;
  padding: 5px 20px 15px 20px;
  border: 1px solid lightgrey;
  border-radius: 3px;
}

input[type=text] {
  width: 100%;
  margin-bottom: 20px;
  padding: 12px;
  border: 1px solid #ccc;
  border-radius: 3px;
}

label {
  margin-bottom: 10px;
  display: block;
}

.icon-container {
  margin-bottom: 20px;
  padding: 7px 0;
  font-size: 24px;
}

.checkout-btn {
  background-color: #4CAF50;
  color: white;
```

```
  padding: 12px;
  margin: 10px 0;
  border: none;
  width: 100%;
  border-radius: 3px;
  cursor: pointer;
  font-size: 17px;
}

.checkout-btn:hover {
  background-color: #45a049;
}


hr {
  border: 1px solid lightgrey;
}

span.price {
  float: right;
  color: grey;
}

/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of ea
ch other instead of next to each other (also change the direction - make the "cart" column go on top) */
@media (max-width: 800px) {
  .row-checkout {
    flex-direction: column-reverse;
  }
  .col-25 {
    margin-bottom: 20px;
  }
}
</style>


<section class="section"style="background:whitesmoke;">
    <div class="container-fluid">
        <div class="row-checkout">
        <?php
        if(isset($_SESSION["uid"])){
            $sql = "SELECT * FROM user_info WHERE user_id='$_SESSION[uid]'";
            $query = mysqli_query($con,$sql);
            $row=mysqli_fetch_array($query);

        echo'
            <div class="col-75">
                <div class="container-checkout">
                <form id="checkout_form" action="checkout_process.php" method="POST" class="was-
validated">
```

```
<div class="row-checkout">

            <div class="col-50">
              <h3>Billing Address</h3>
              <label for="fname"><i class="fa fa-user" ></i> Full Name</label>
              <input type="text" id="fname" class="form-control" name="firstname" pattern="^[a-zA-
Z ]+$"  value="'.$row["first_name"].' '.$row["last_name"].'">
              <label for="email"><i class="fa fa-envelope"></i> Email</label>
              <input type="text" id="email" name="email" class="form-control" pattern="^[_a-z0-9-
]+(\.[_a-z0-9-]+)*@[a-z0-9]+(\.[a-z]{2,4})$" value="'.$row["email"].'" required>
              <label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
              <input type="text" id="adr" name="address" class="form-
control" value="'.$row["address1"].'" required>
              <label for="city"><i class="fa fa-institution"></i> City</label>
              <input type="text" id="city" name="city" class="form-
control" value="'.$row["address2"].'" pattern="^[a-zA-Z ]+$" required>

              <div class="row">
              <div class="col-50">
                <label for="state">State</label>
                <input type="text" id="state" name="state" class="form-control" pattern="^[a-zA-
Z ]+$" required>
              </div>
              <div class="col-50">
                <label for="zip">Zip</label>
                <input type="text" id="zip" name="zip" class="form-control" pattern="^[0-9]{6}(?:-[0-
9]{4})?$" required>
              </div>
              </div>
            </div>


            <div class="col-50">
              <h3>Payment</h3>
              <label for="fname">Accepted Cards</label>
              <div class="icon-container">
              <i class="fa fa-cc-visa" style="color:navy;"></i>
              <i class="fa fa-cc-amex" style="color:blue;"></i>
              <i class="fa fa-cc-mastercard" style="color:red;"></i>
              <i class="fa fa-cc-discover" style="color:orange;"></i>
              </div>


              <label for="cname">Name on Card</label>
              <input type="text" id="cname" name="cardname" class="form-control" pattern="^[a-zA-
Z ]+$" required>

              <div class="form-group" id="card-number-field">
              <label for="cardNumber">Card Number</label>
              <input type="text" class="form-
control" id="cardNumber" name="cardNumber" required>
```

```
</div>
                <label for="expdate">Exp Date</label>
                <input type="text" id="expdate" name="expdate" class="form-control" pattern="^((0[1-
9])|(1[0-2]))\/(\d{2})$" placeholder="12/22"required>


                <div class="row">

                <div class="col-50">
                   <div class="form-group CVV">
                      <label for="cvv">CVV</label>
                      <input type="text" class="form-control" name="cvv" id="cvv" required>
                </div>
                </div>
              </div>
              </div>
              </div>
              <label><input type="CHECKBOX" name="q" class="roomselect" value="conform" require
d> Shipping address same as billing
              </label>';
              $i=1;
              $total=0;
              $total_count=$_POST['total_count'];
              while($i<=$total_count){
                 $item_name_ = $_POST['item_name_'.$i];
                 $amount_ = $_POST['amount_'.$i];
                 $quantity_ = $_POST['quantity_'.$i];
                 $total=$total+$amount_ ;
                 $sql = "SELECT product_id FROM products WHERE product_title='$item_name_'";
                 $query = mysqli_query($con,$sql);
                 $row=mysqli_fetch_array($query);
                 $product_id=$row["product_id"];
                 echo "
                 <input type='hidden' name='prod_id_$i' value='$product_id'>
                 <input type='hidden' name='prod_price_$i' value='$amount_'>
                 <input type='hidden' name='prod_qty_$i' value='$quantity_'>
                 ";
                 $i++;
              }

           echo'
           <input type="hidden" name="total_count" value="'.$total_count.'">
              <input type="hidden" name="total_price" value="'.$total.'">

              <input type="submit" id="submit" value="Continue to checkout" class="checkout-btn">
           </form>
           </div>
         </div>
         ';
      }else{
```

```
 echo"<script>window.location.href = 'cart.php'</script>";
}
      ?>

         <div class="col-25">
            <div class="container-checkout">

            <?php
            if (isset($_POST["cmd"])) {

               $user_id = $_POST['custom'];


               $i=1;
               echo
               "
               <h4>Cart
               <span class='price' style='color:black'>
               <i class='fa fa-shopping-cart'></i>
               <b>$total_count</b>
               </span>
            </h4>

               <table class='table table-condensed'>
               <thead><tr>
               <th >no</th>
               <th >product title</th>
               <th >   qty </th>
               <th >   amount</th></tr>
               </thead>
               <tbody>
               ";
               $total=0;
               while($i<=$total_count){
                  $item_name_ = $_POST['item_name_'.$i];

                  $item_number_ = $_POST['item_number_'.$i];

                  $amount_ = $_POST['amount_'.$i];

                  $quantity_ = $_POST['quantity_'.$i];
                  $total=$total+$amount_ ;
                  $sql = "SELECT product_id FROM products WHERE product_title='$item_name_'";
                  $query = mysqli_query($con,$sql);
                  $row=mysqli_fetch_array($query);
                  $product_id=$row["product_id"];

                  echo "
```

```
<tr><td><p>$item_number_</p></td><td><p>$item_name_</p></td><td ><p>$quantity_</p></td><td
><p>$amount_</p></td></tr>";

                $i++;
              }

            echo"

            </tbody>
            </table>
            <hr>

            <h3>total<span class='price' style='color:black'><b>Rs $total</b></span></h3>";

            }
            ?>
            </div>
          </div>
        </div>
    </div>
</section>

<?php
include "footer.php";
?>
```

Register.php

```
<?php
session_start();
include "db.php";
if (isset($_POST["f_name"])) {

    $f_name = $_POST["f_name"];
    $l_name = $_POST["l_name"];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $repassword = $_POST['repassword'];
    $mobile = $_POST['mobile'];
    $address1 = $_POST['address1'];
    $address2 = $_POST['address2'];
    $name = "/^[a-zA-Z ]+$/";
    $emailValidation = "/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9]+(\.[a-z]{2,4})$/";
    $number = "/^[0-9]+$/";

if(empty($f_name) || empty($l_name) || empty($email) || empty($password) || empty($repassword) ||
    empty($mobile) || empty($address1) || empty($address2)){
```

```
 echo "
          <div class='alert alert-warning'>


<a href='#' class='close' data-dismiss='alert' aria-
label='close'>&times;</a><b>PLease Fill all fields..!</b>
          </div>
        ";
        exit();
    } else {
        if(!preg_match($name,$f_name)){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>this $f_name is not valid..!</b>
            </div>
        ";
        exit();
    }
    if(!preg_match($name,$l_name)){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>this $l_name is not valid..!</b>
            </div>
        ";
        exit();
    }
    if(!preg_match($emailValidation,$email)){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>this $email is not valid..!</b>
            </div>
        ";
        exit();
    }
    if(strlen($password) < 9 ){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>Password is weak</b>
            </div>
        ";
        exit();
    }
    if(strlen($repassword) < 9 ){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
```

```
<b>Password is weak</b>
            </div>
        ";
        exit();
    }


if($password != $repassword){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>password is not same</b>
            </div>
        ";
    }
    if(!preg_match($number,$mobile)){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>Mobile number $mobile is not valid</b>
            </div>
        ";
        exit();
    }
    if(!(strlen($mobile) == 10)){
        echo "
            <div class='alert alert-warning'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>Mobile number must be 10 digit</b>
            </div>
        ";
        exit();
    }
    //existing email address in our database
    $sql = "SELECT user_id FROM user_info WHERE email = '$email' LIMIT 1" ;
    $check_query = mysqli_query($con,$sql);
    $count_email = mysqli_num_rows($check_query);
    if($count_email > 0){
        echo "
            <div class='alert alert-danger'>
                <a href='#' class='close' data-dismiss='alert' aria-label='close'>&times;</a>
                <b>Email Address is already available Try Another email address</b>
            </div>
        ";
        exit();
    } else {

        $sql = "INSERT INTO `user_info`
        (`user_id`, `first_name`, `last_name`, `email`,
        `password`, `mobile`, `address1`, `address2`)
```

```
VALUES (NULL, '$f_name', '$l_name', '$email',
      '$password', '$mobile', '$address1', '$address2')";
      $run_query = mysqli_query($con,$sql);
      $_SESSION["uid"] = mysqli_insert_id($con);
      $_SESSION["name"] = $f_name;
      $ip_add = getenv("REMOTE_ADDR");
      $sql = "UPDATE cart SET user_id = '$_SESSION[uid]' WHERE ip_add='$ip_add' AND user_id = -
1";


if(mysqli_query($con,$sql)){
          echo "register_success";
          echo "<script> location.href='store.php'; </script>";
          exit;
      }
   }
   }

}


?>
```

Login.php

```
<?php
include "db.php";

session_start();

#Login script is begin here
#If user given credential matches successfully with the data available in database then we will echo string
login_success
#login_success string will go back to called Anonymous funtion $("#login").click()

if(isset($_POST["email"]) && isset($_POST["password"])){
   $email = mysqli_real_escape_string($con,$_POST["email"]);
   $password = $_POST["password"];
   $sql = "SELECT * FROM user_info WHERE email = '$email' AND password = '$password'";
   $run_query = mysqli_query($con,$sql);
   $count = mysqli_num_rows($run_query);
   $row = mysqli_fetch_array($run_query);
      $_SESSION["uid"] = $row["user_id"];
      $_SESSION["name"] = $row["first_name"];
      $ip_add = getenv("REMOTE_ADDR");
      //we have created a cookie in login_form.php page so if that cookie is available means user is not log
in

   //if user record is available in database then $count will be equal to 1
```

```
 if($count == 1){

        if (isset($_COOKIE["product_list"])) {
           $p_list = stripcslashes($_COOKIE["product_list"]);
           //here we are decoding stored json product list cookie to normal array
           $product_list = json_decode($p_list,true);
           for ($i=0; $i < count($product_list); $i++) {
              //After getting user id from database here we are checking user cart item if there is already pr
oduct is listed or not
              $verify_cart = "SELECT id FROM cart WHERE user_id = $_SESSION[uid] AND p_id = ".
$product_list[$i];
              $result  = mysqli_query($con,$verify_cart);

if(mysqli_num_rows($result) < 1){
                 //if user is adding first time product into cart we will update user_id into database table wit
h valid id
                 $update_cart = "UPDATE cart SET user_id = '$_SESSION[uid]' WHERE ip_add = '$ip_a
dd' AND user_id = -1";
                 mysqli_query($con,$update_cart);
              }else{
                 //if already that product is available into database table we will delete that record
                 $delete_existing_product = "DELETE FROM cart WHERE user_id = -
1 AND ip_add = '$ip_add' AND p_id = ".$product_list[$i];
                 mysqli_query($con,$delete_existing_product);
              }
           }
           //here we are destroying user cookie
           setcookie("product_list","",strtotime("-1 day"),"/");
           //if user is logging from after cart page we will send cart_login
           echo "cart_login";


           exit();

        }
        //if user is login from page we will send login_success
        echo "login_success";
        $BackToMyPage = $_SERVER['HTTP_REFERER'];
           if(!isset($BackToMyPage)) {
              header('Location: '.$BackToMyPage);
              echo"<script type='text/javascript'>

              </script>";
           } else {
              header('Location: index.php'); // default page
           }


        exit;
```

```
}else{
        $email = mysqli_real_escape_string($con,$_POST["email"]);
        $password =md5($_POST["password"]) ;
        $sql = "SELECT * FROM admin_info WHERE admin_email = '$email' AND admin_password
= '$password'";
        $run_query = mysqli_query($con,$sql);
        $count = mysqli_num_rows($run_query);

      //if user record is available in database then $count will be equal to 1
      if($count == 1){
        $row = mysqli_fetch_array($run_query);
        $_SESSION["uid"] = $row["admin_id"];
        $_SESSION["name"] = $row["admin_name"];
        $ip_add = getenv("REMOTE_ADDR");


//we have created a cookie in login_form.php page so if that cookie is available means user is not login

            //if user is login from page we will send login_success
            echo "login_success";

            echo "<script> location.href='admin/add_product.php'; </script>";
            exit;

        }else{
            echo "<span style='color:red;'>Please register before login..!</span>";
            exit();
        }


}


}

?>
```

Chapter 5

SNAPSHOTS

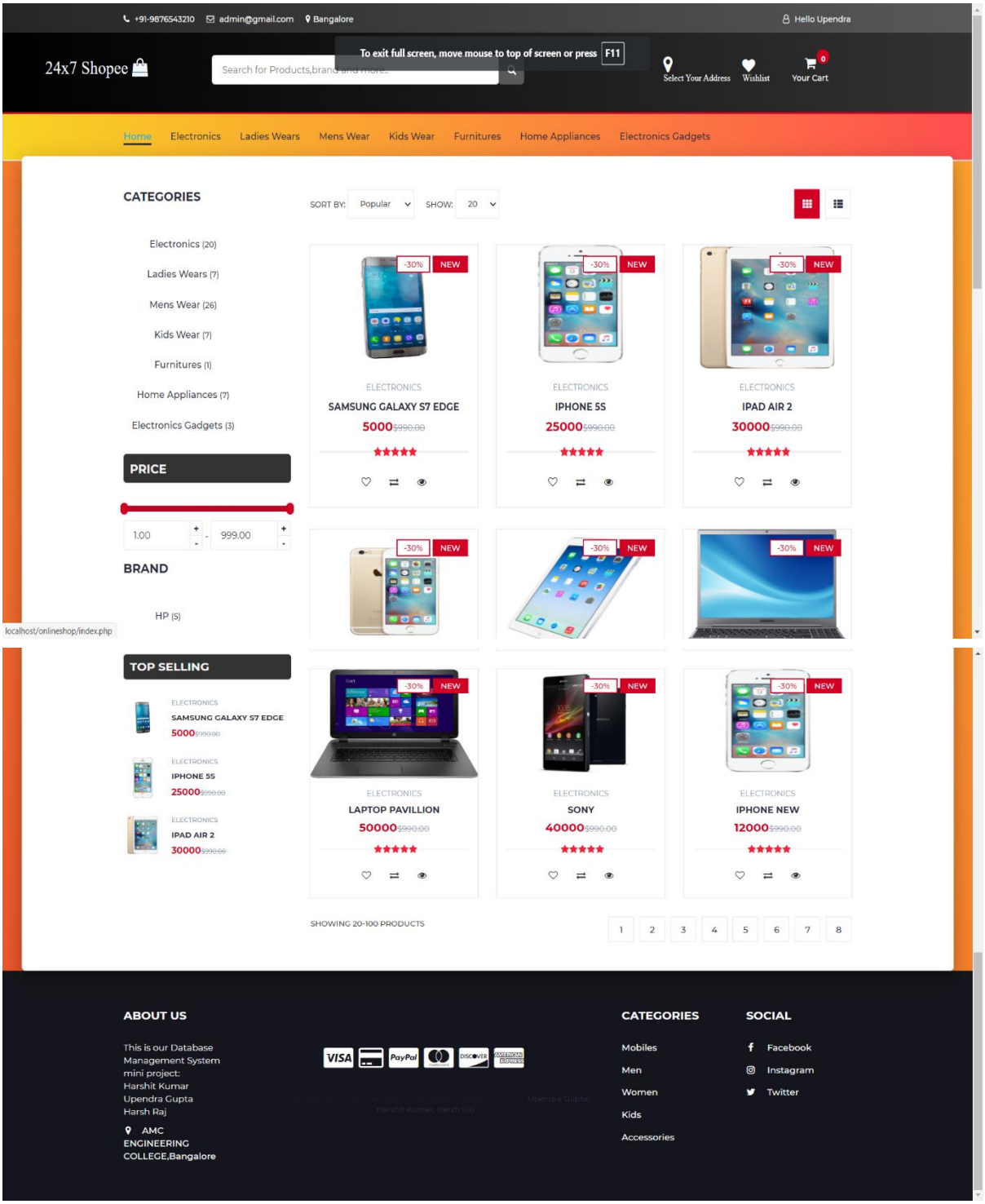

Figure 5.1 : Homepage

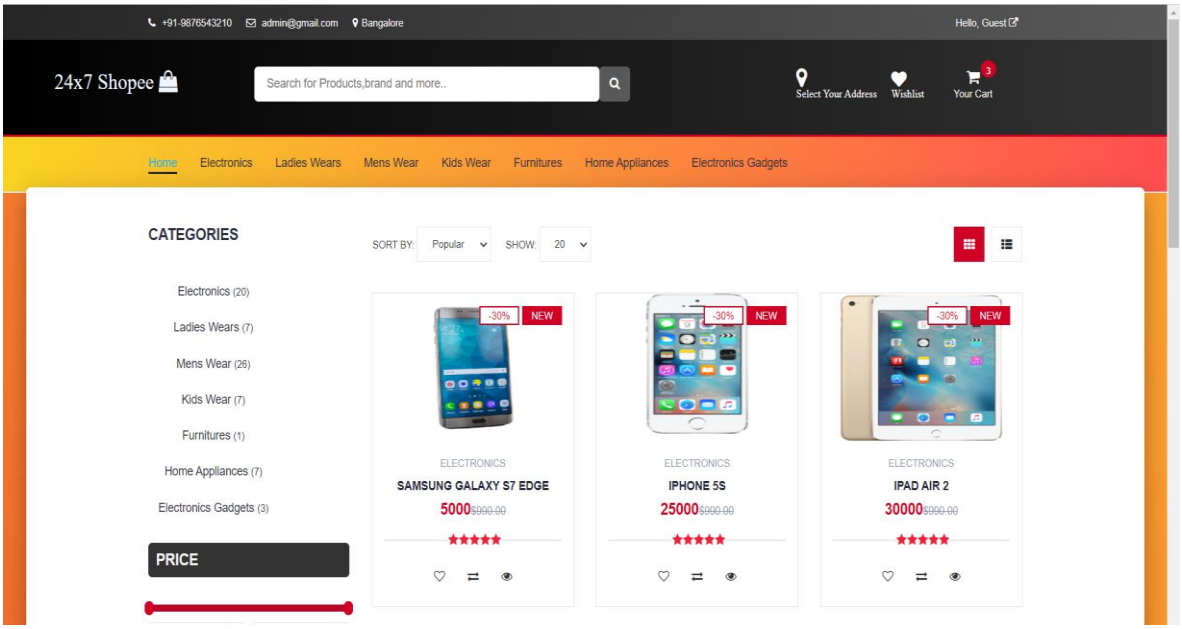

Figure 5.2 : Electronic product page

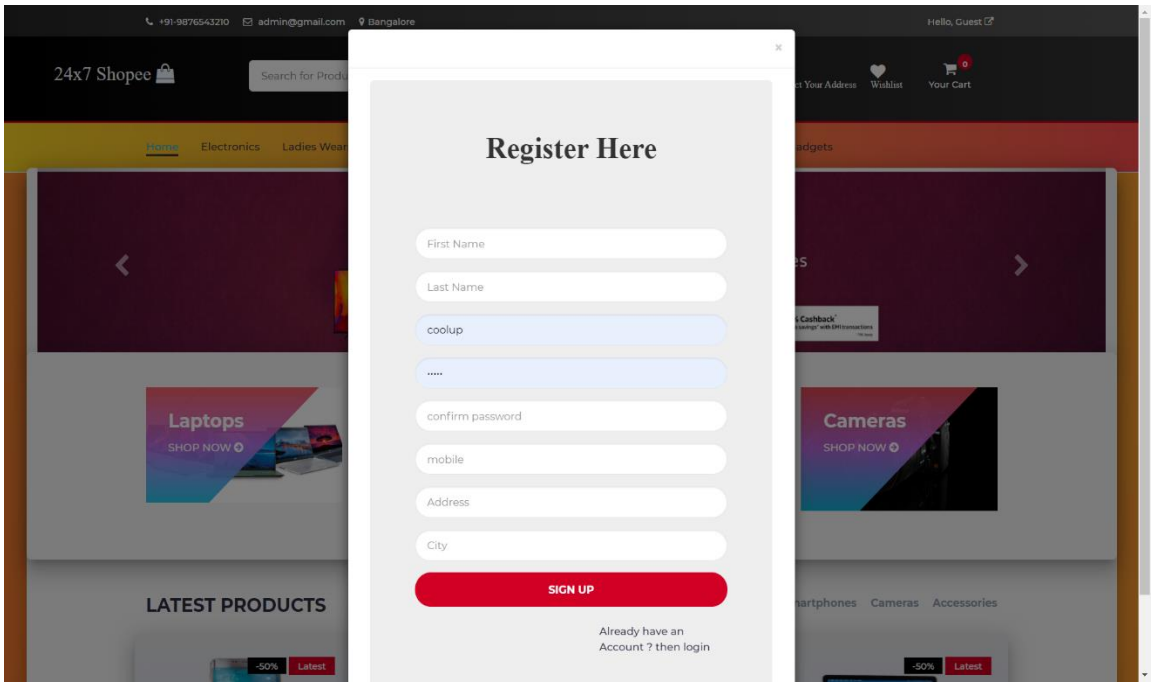

Figure 5.3 : Sign-up page

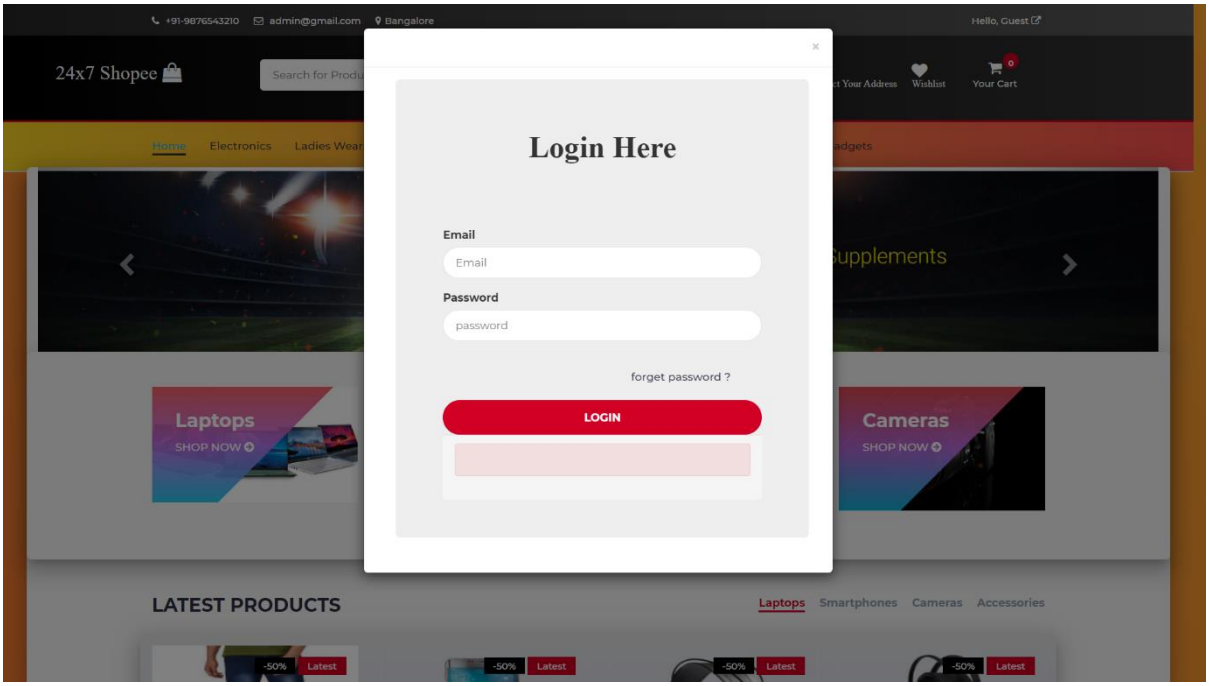

Figure 5.4 : Login page

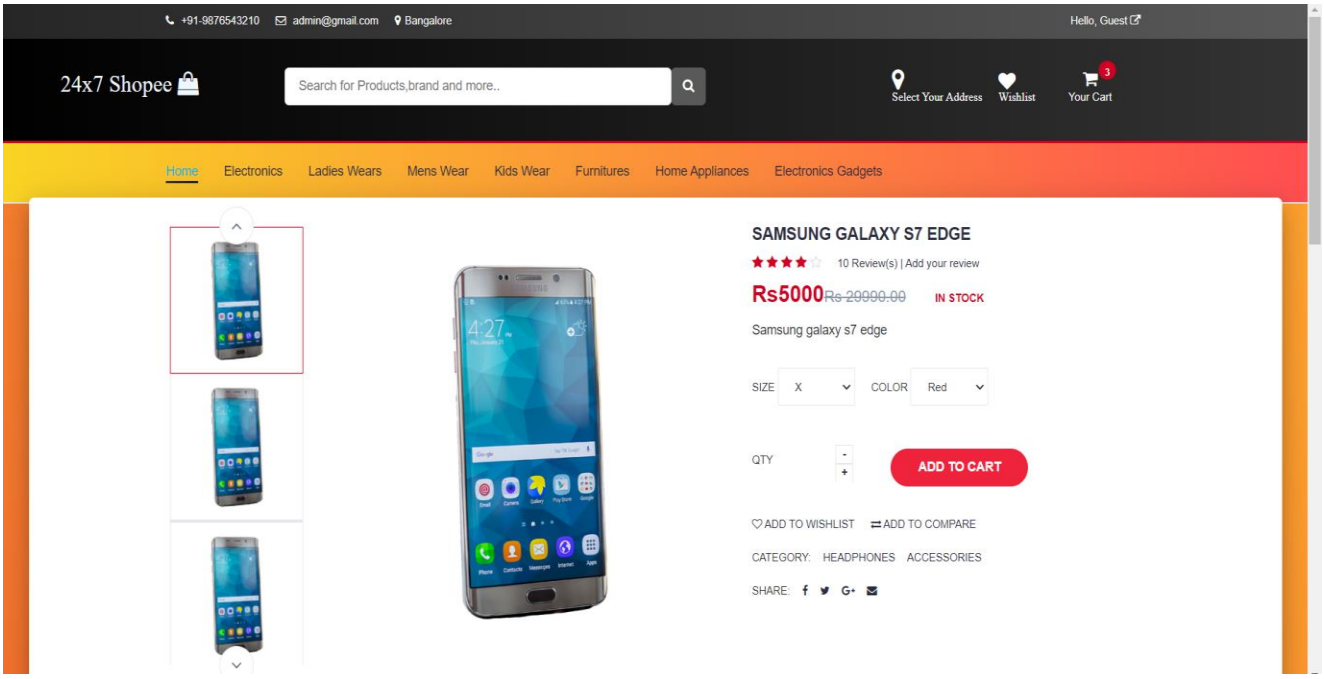

Figure 5.5 : Product Page

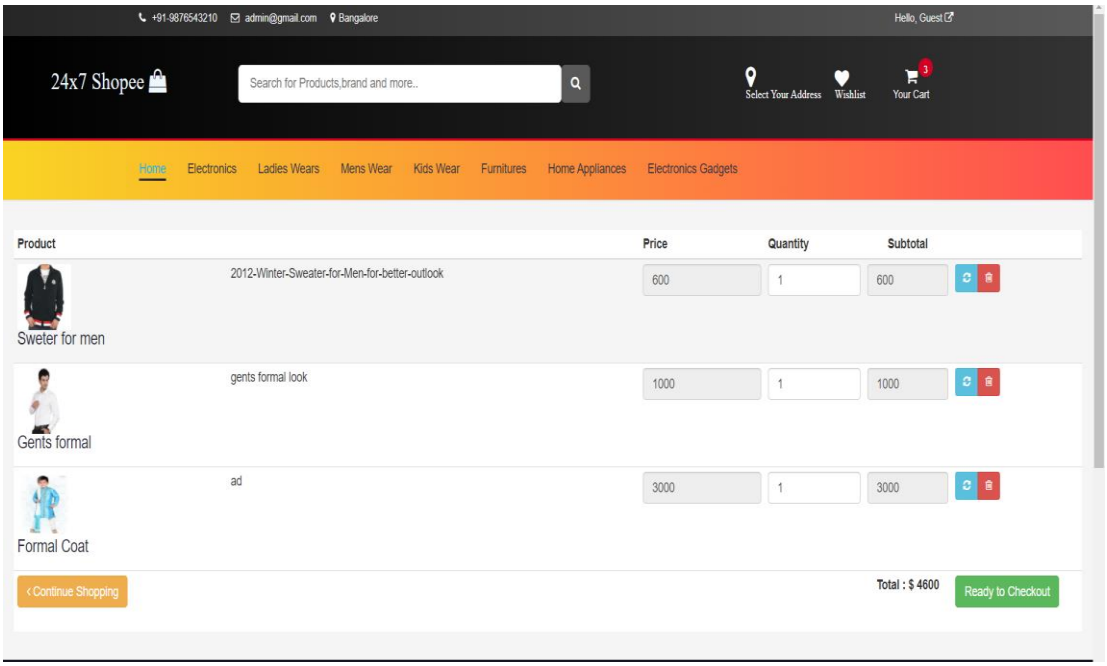

Figure 5.6 : Cart Page

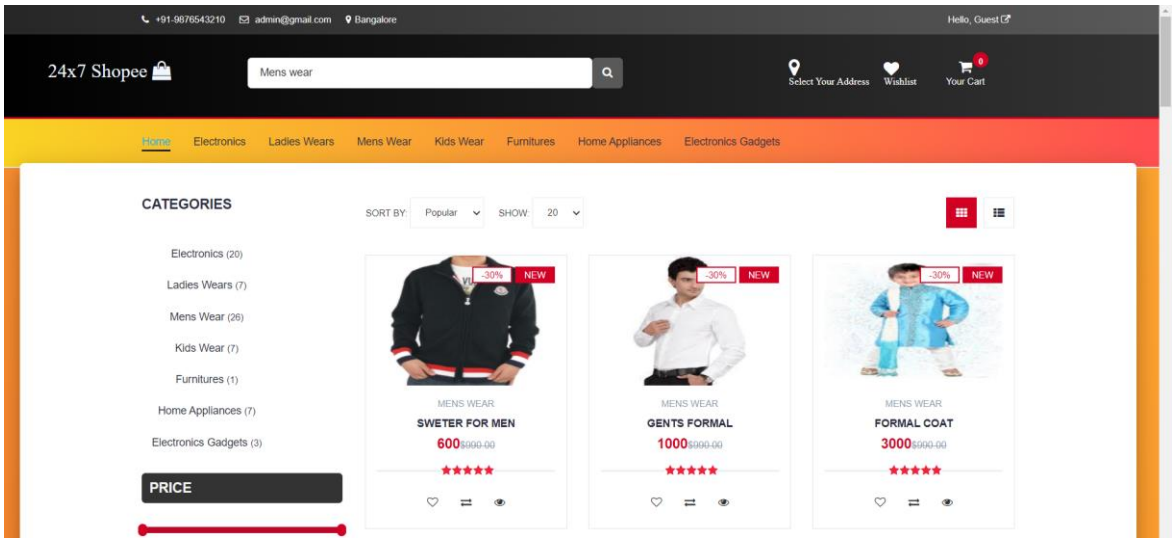

Figure 5.5 : Search Page

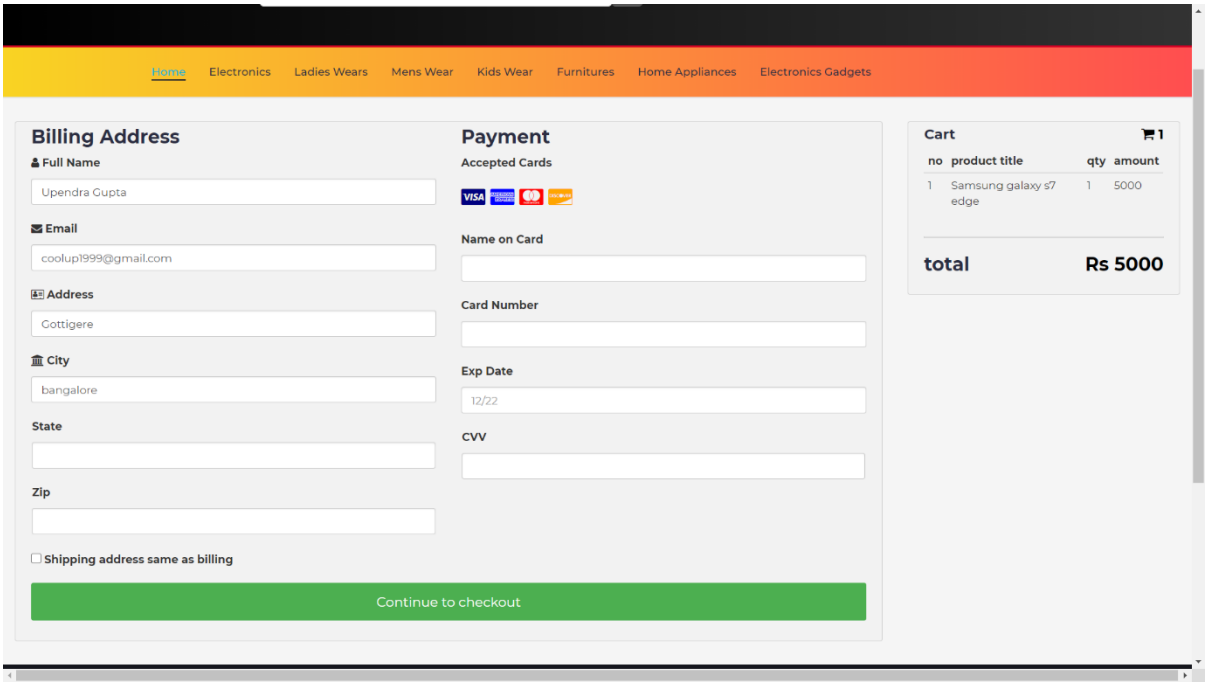

Figure 5.6 : Payment Page

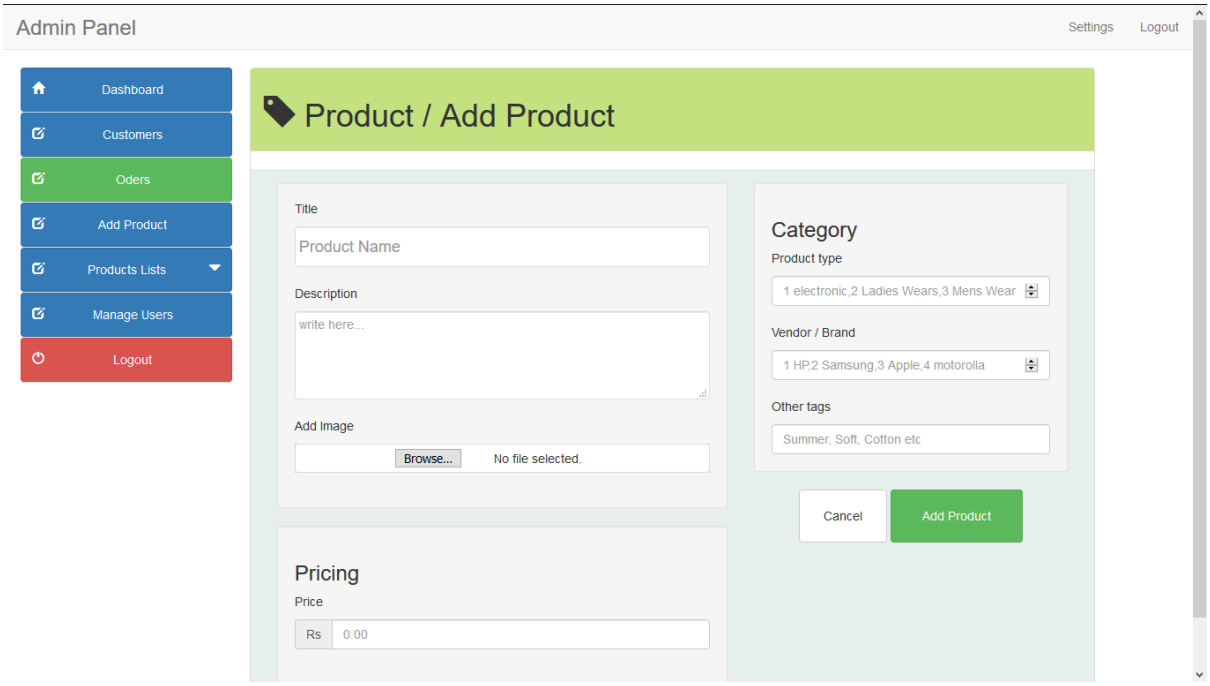

Figure 5.7 : Add Product Page

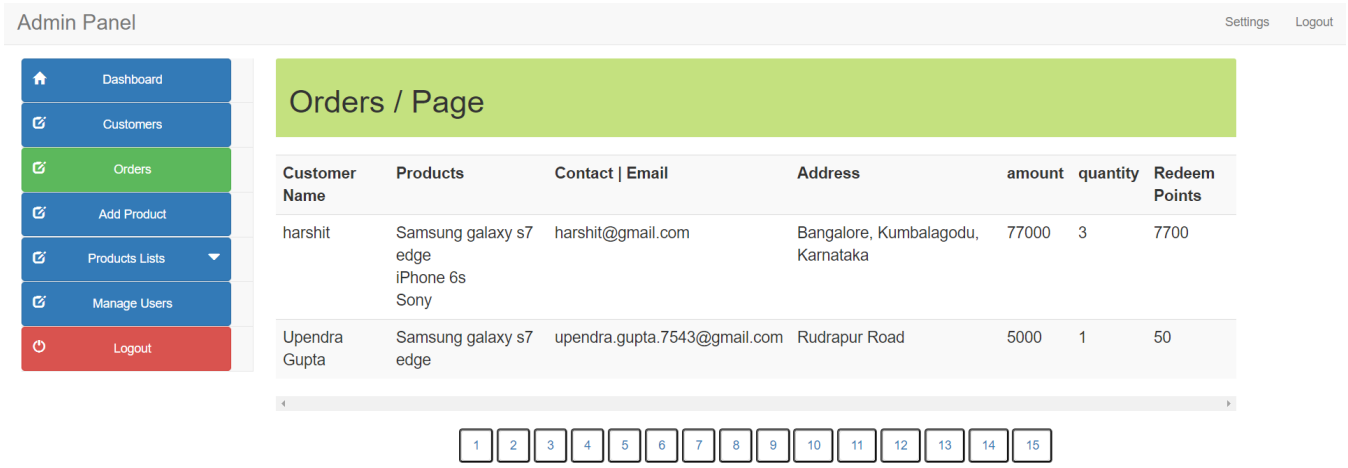

Figure 5.8 : Order Page

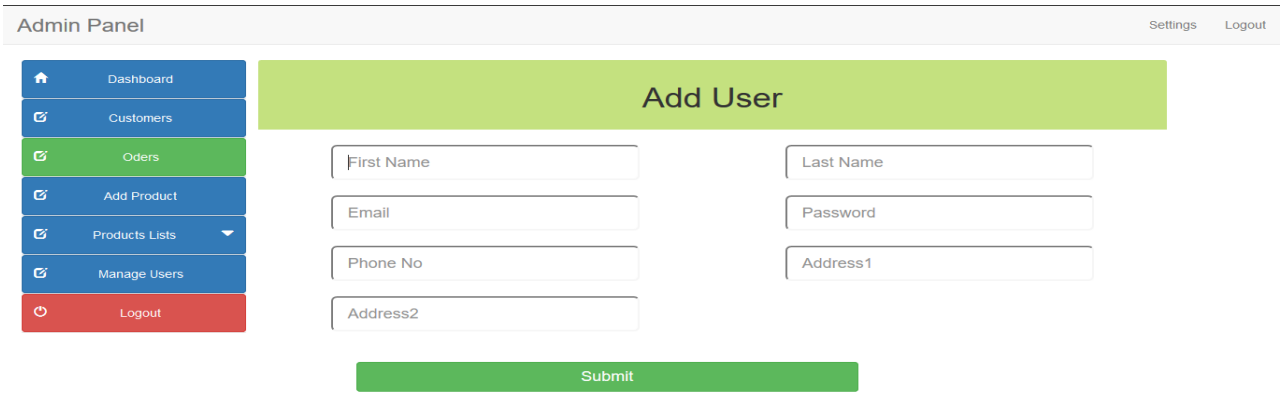

Figure 5.9 : Add User Page

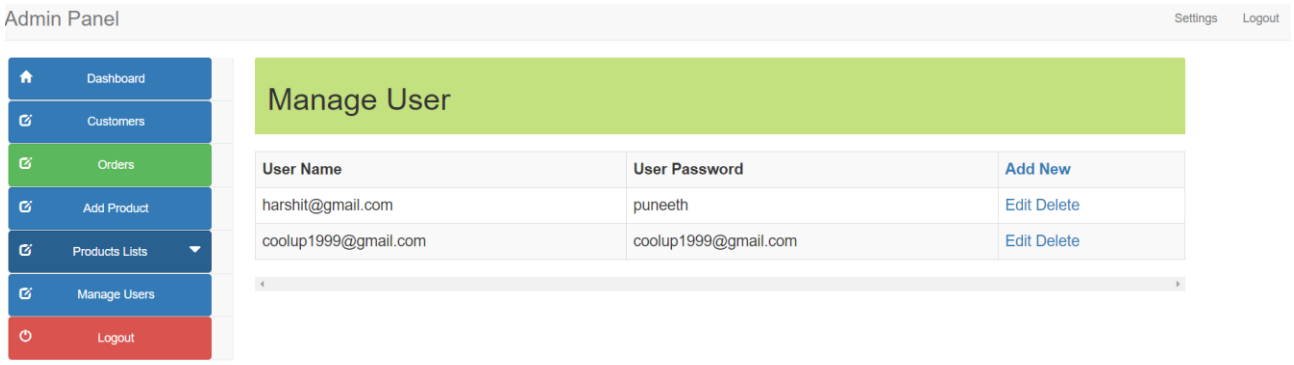

Figure 5.10 : Manage user Page

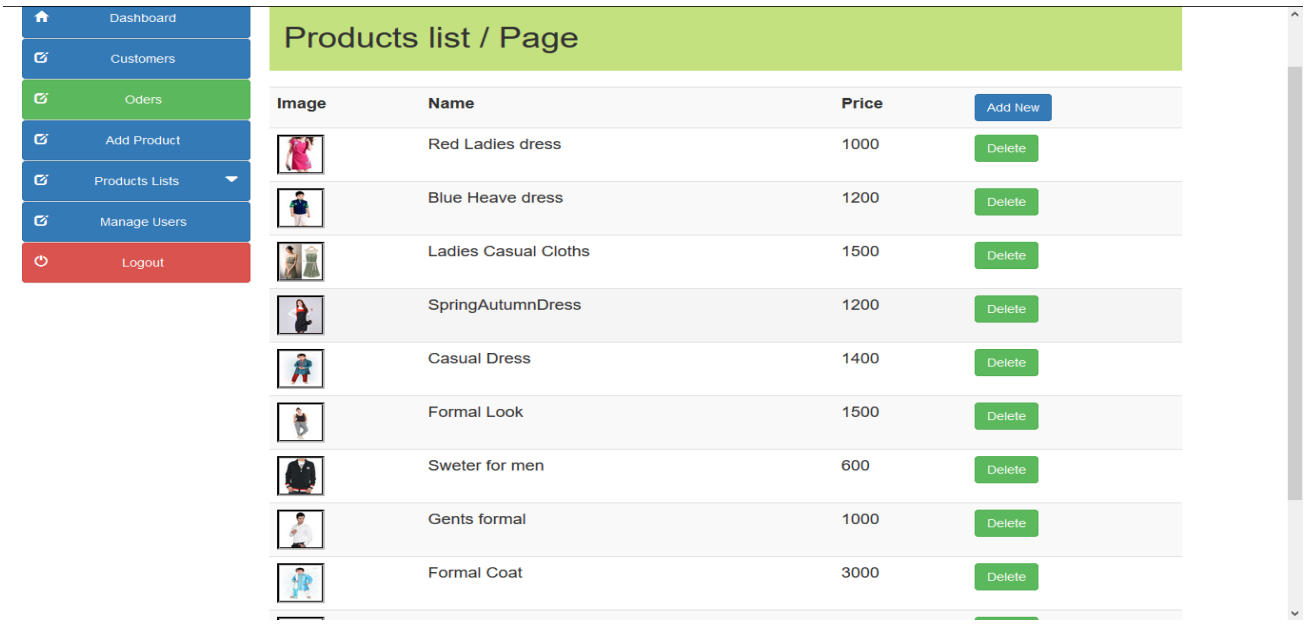

Figure 5.11 : Product list Page

Chapter 6

CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of PHP my admin and no some extent Windows Application and SQL Server, but also about all handling procedure related with "ONLINE SHOPPING DATABASE SYSTEM". It also provides knowledge about the latest technology used in developing standalone application and client server technology that will be great demand in future.

The project entitled Online shopping system was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application and an android application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using HTML & CSS, usage of responsive templates, designing of android applications, and management of database using MySQL. The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project. This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing

REFERENCES

[1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014,McGraw Hill.

[3] Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, Mc-GrawHill, 2013. [4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

WEB URL'S:

https://www.w3schools.com/
https://www.tutorialspoint.com/
http://stackoverflow.com/