Multi-vendor e-commerce system

Programming in php

Al Hussein Technical University

February-2021

Supervised by: Eng. Salameh Yassin

Student name: Noor alhuda Al-juhani

ABSTRACT

Electronic Commerce is process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products.

Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990's has taken a great leap in the world of computers, but the fact that has hindered the growth of e-commerce is security.
Security is the challenge facing e-commerce today & there is still a lot of advancement made in the field of security.

The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC.
For increasing the use of e-commerce in developing countries the B2B e-commerce is implemented for improving access to global markets for firms in developing countries.

For a developing country advancement in the field of e-commerce is essential.
Electronic commerce is a term for any type of business, or commercial transaction,
that involves the transfer of information across the Internet. It covers a range of different types of businesses.

Table of contents

Introduction

1.1Preamble:

With the increasing growth in the electronic commerce sector and the increase in the number of its users, it has become important to think about the buyer and how to provide him with comfort and safety during the electronic purchase process.

Therefore, we had the idea of multiple sellers in one website, which reduces the user effort and time and also provides the sellers with a greater opportunity to obtain a number of new users.

The multi-vendor system in electronic commerce has had a positive effect on all, whether sellers or users.

So far, my project will be a multi-vendor trading system to handle the e-shopping process

Within any company and providing a very clever way to showcase the goods from the traditional

Attractive new ways using web application development languages

(PHP, HTML, CSS, JavaScript, Bootstrap, MySQL).

1.2 Project Motivation:

This system will provide a very powerful and new smart technology in the field of electronic commerce, using the latest programming languages and linking them in an excellent way to achieve the desired result of the project, and it also provides a unique and enjoyable experience for all elements, whether vendors or customers, to share the required information between them all based on their privileges.

1.3 Project goals:

The aim of this system is to facilitate the buying and selling process

And make it an enjoyable and safe process for both buyers and sellers and attract the largest possible number of buyers and provide an enjoyable experience that helps reduce the time and effort spent by both parties.

Study existing systems:

The starting point for many projects is often a similar or an existing system. Sometimes, comparable

products and systems contain working versions of good ideas for solving user problems. We can Save the time lost in reinventing the wheel by looking at systems already on the market, whether they are systems installed at the user's site or products made by rival organizations.

Software development life cycle

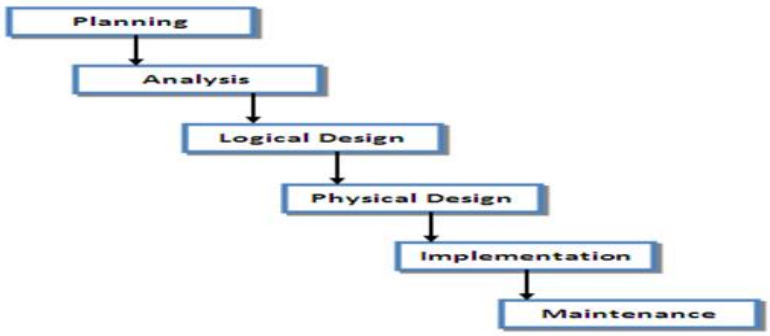


Figure (1): Waterfall SDLC.

-Project Initiation and Planning:

In this section I define project feasibility, identify project and identify its scope.

-Project Analysis:

In this phase, I will understand system needs and requirements, then I structure these requirements; I mean providing description for system activities and processes.

-Project Design:

The main phase activities are to identify Database relations, transfer these relations to ER-Diagram that show all overall system relations then transform ER-Diagram to physical Tables, after that I will design system Forms and Reports. Implementation: This phase includes Coding, installations and documentation completion.

-Testing:

During this phase this system will be tested with real data to check about every function and how it works as a live online test. Maintenance: This is the final phase of the SDLC to check the Information System is systematically repaired and improved.

-Summary:

Finally based on the previous titles, this chapter define the SDLC module I worked on it and include my system requirements and its analysis with full description about the functionality of the system.

System Design

Important figures that describe my website process, it will include site map of the system, entity relation diagram (ERD) , class diagrams and Mock ups.

- Site map of the system

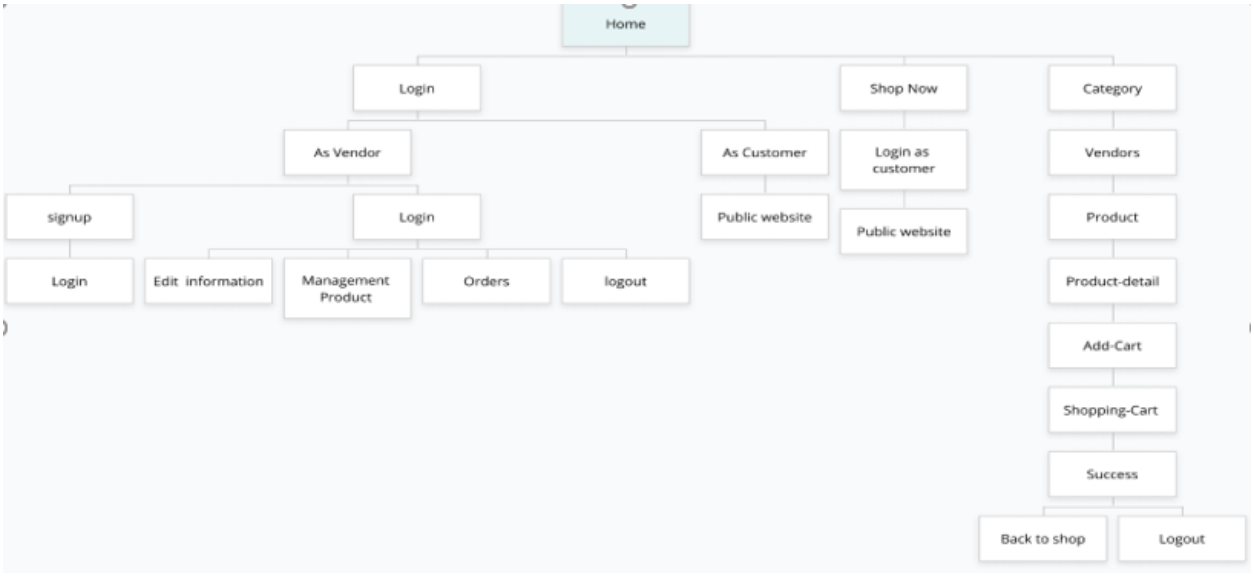


Figure (2): site map

-ERD

Figure (3): ERD.

-Class diagram

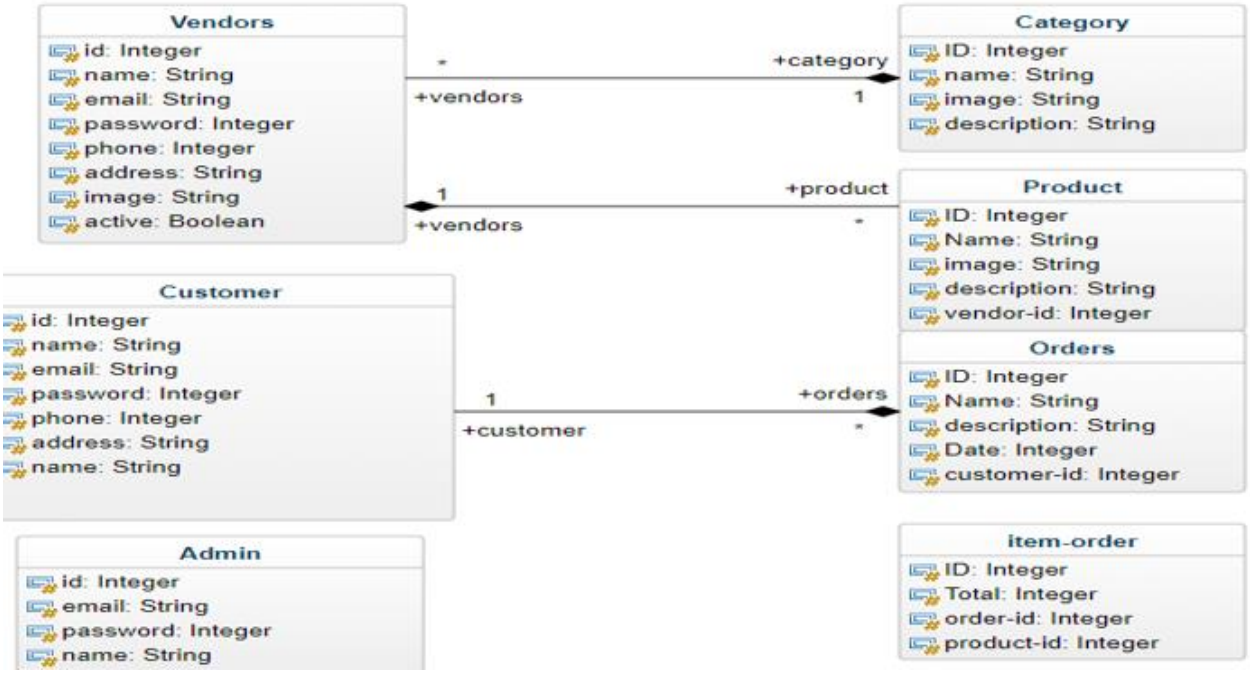
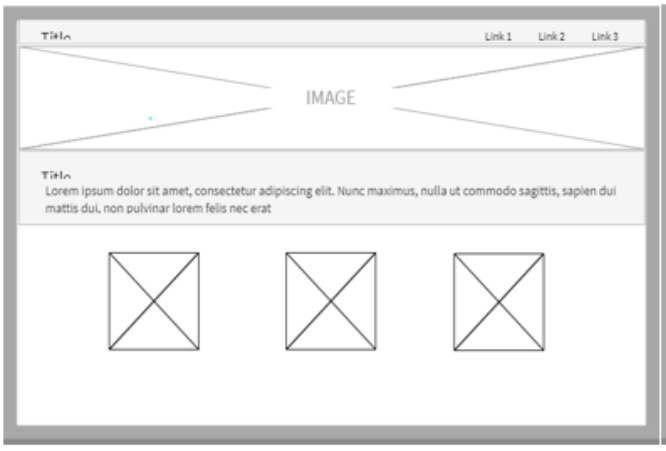
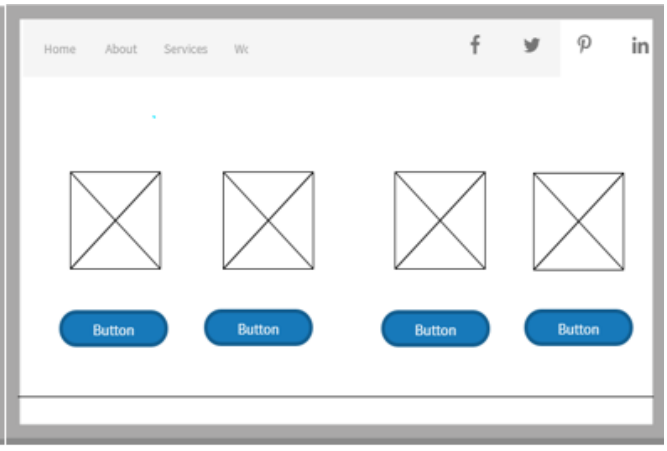


Figure (4): Class diagram.

-Mock-ups



-1-



-2-

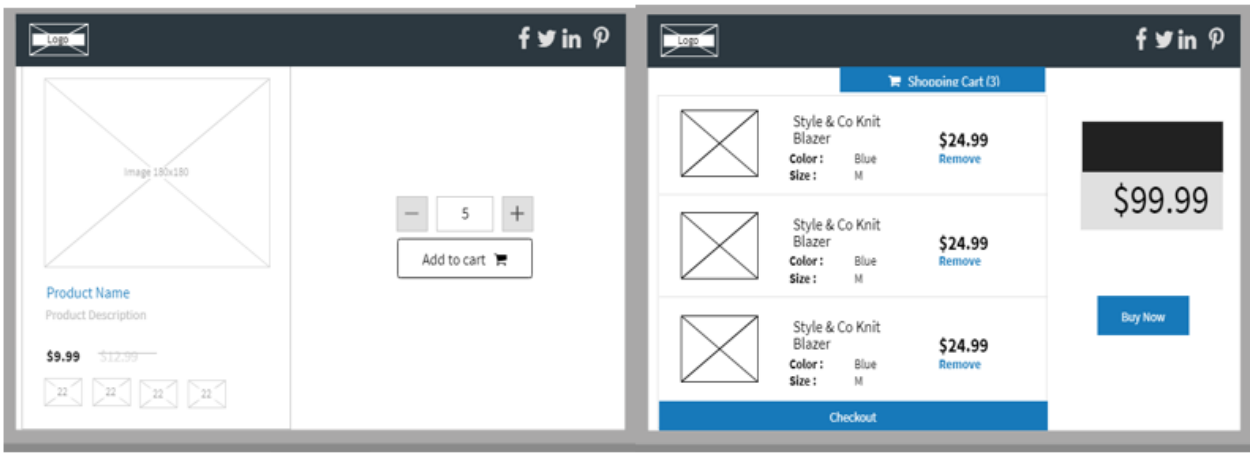


Figure (5): Mocha-ups.

Summary

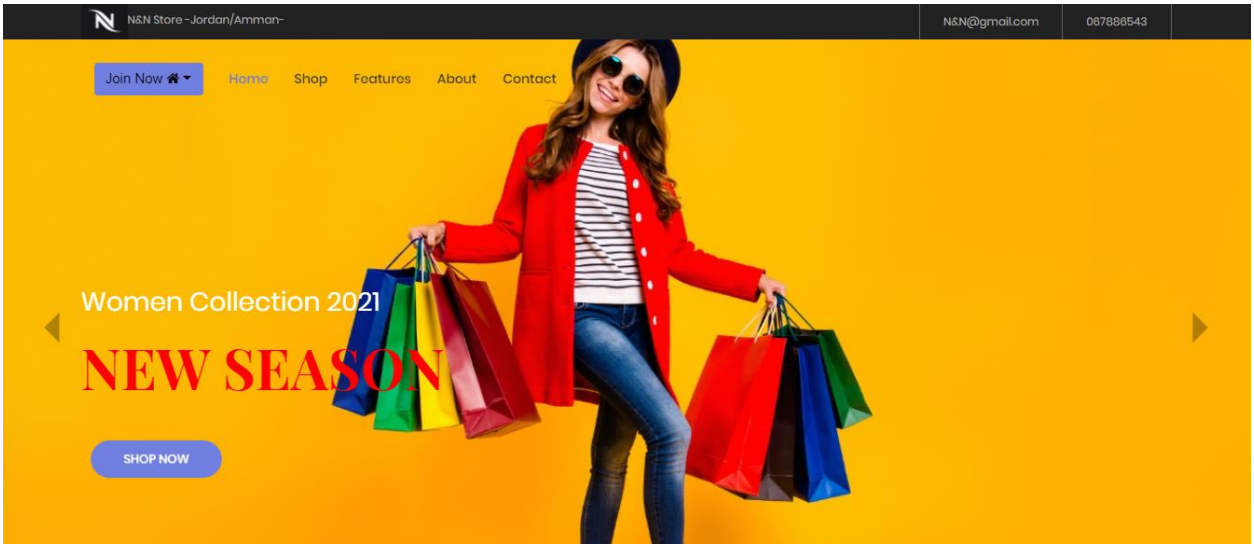
At final, this defines my website functionality and how the data transfer inside it the methodology of the system, that design the site map, entity relationship diagram for database ,class diagram for database relations and Mock-ups.

Graphical User Interface Implementation

Public site:

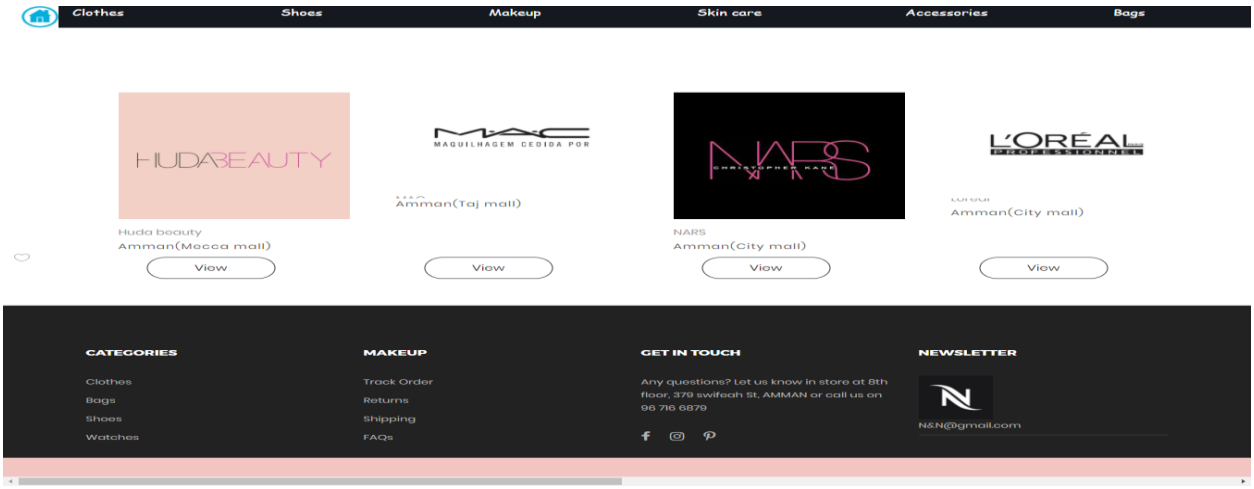
1.Home page

In this page customer or vendor can see all category and some of product in website and login.



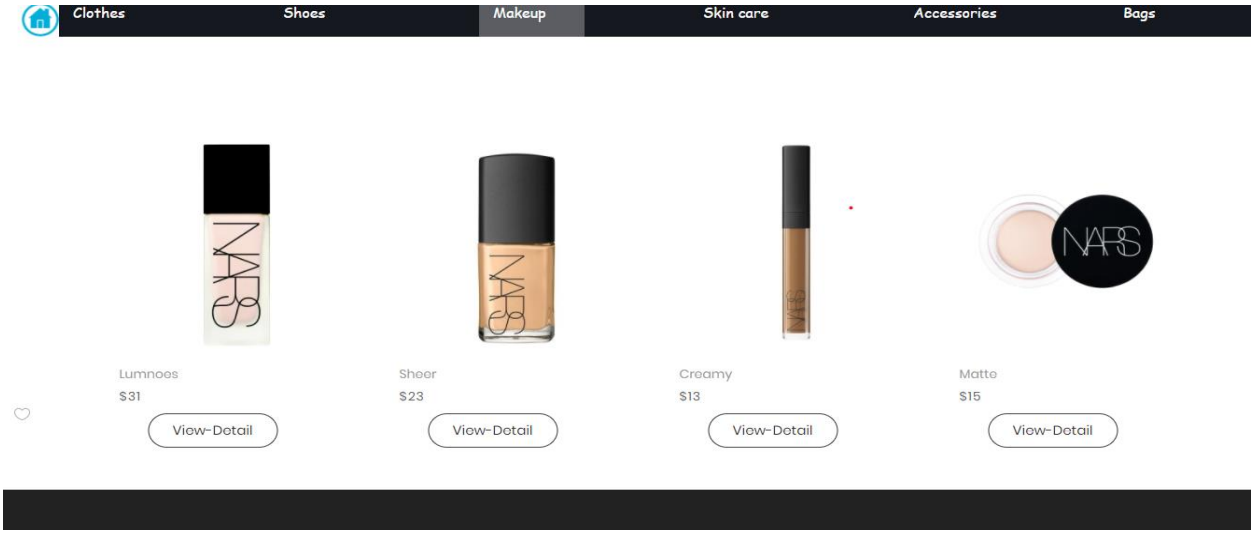
2-Vendor page

In this page you can see all vendor of the category.



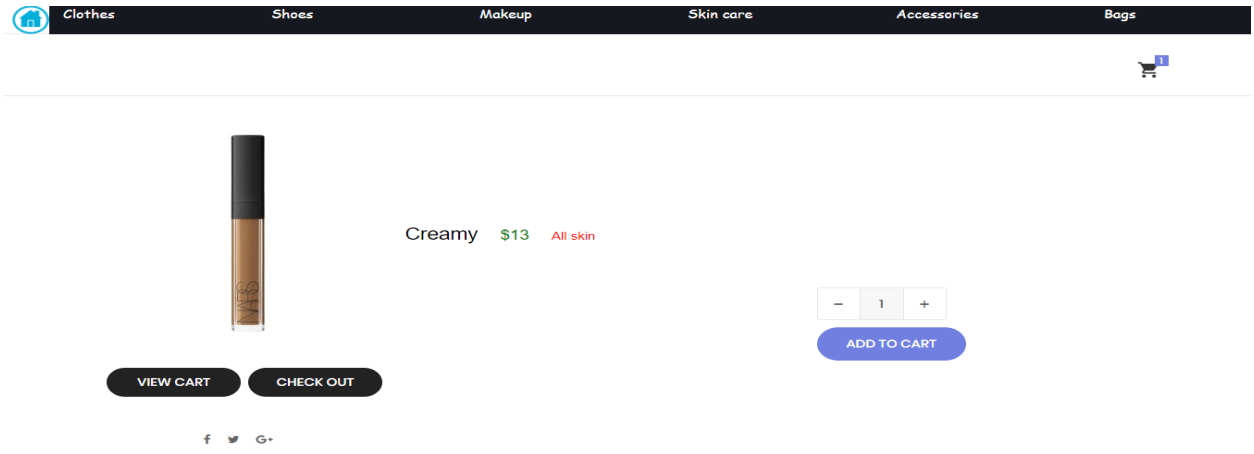
3-Product page

In this page you can see all product of the vendor.



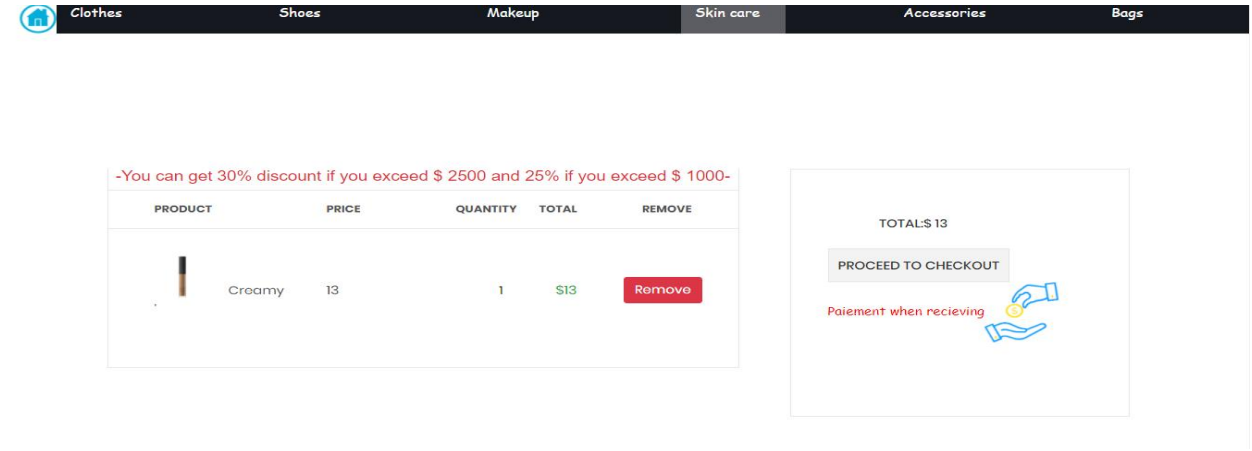
4-Proudect detail page

In this page you are see all detail of the product and add to cart.



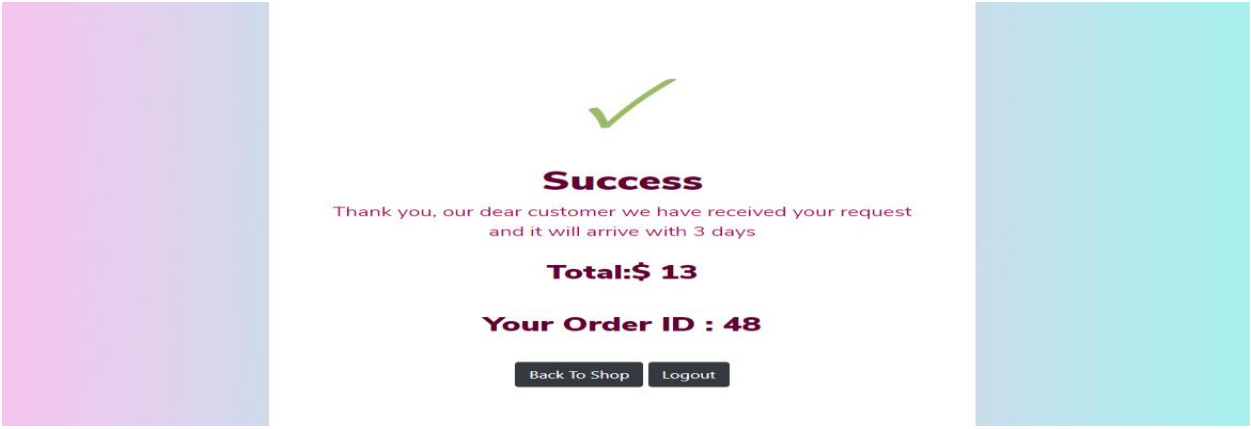
5-Shopping cart

In this page you can see all product in your cart with the detail and total.



6-Success page

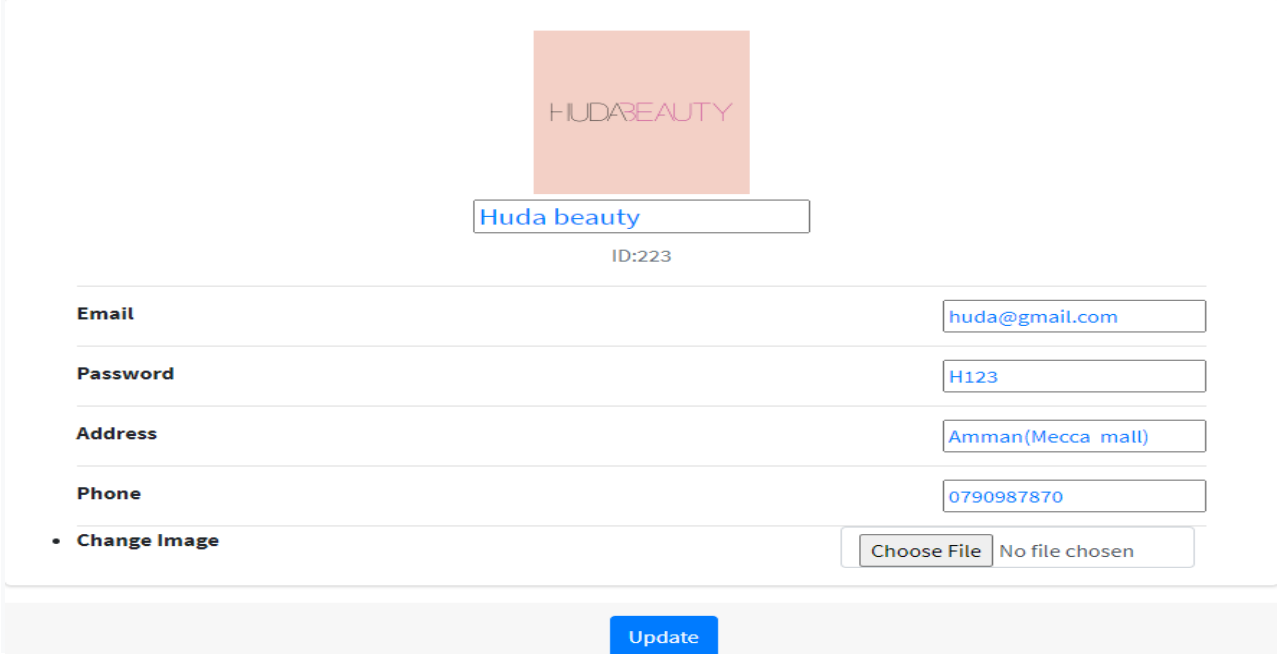
This is the end page in the website show the end total and your order id.



-Vendors dashboard

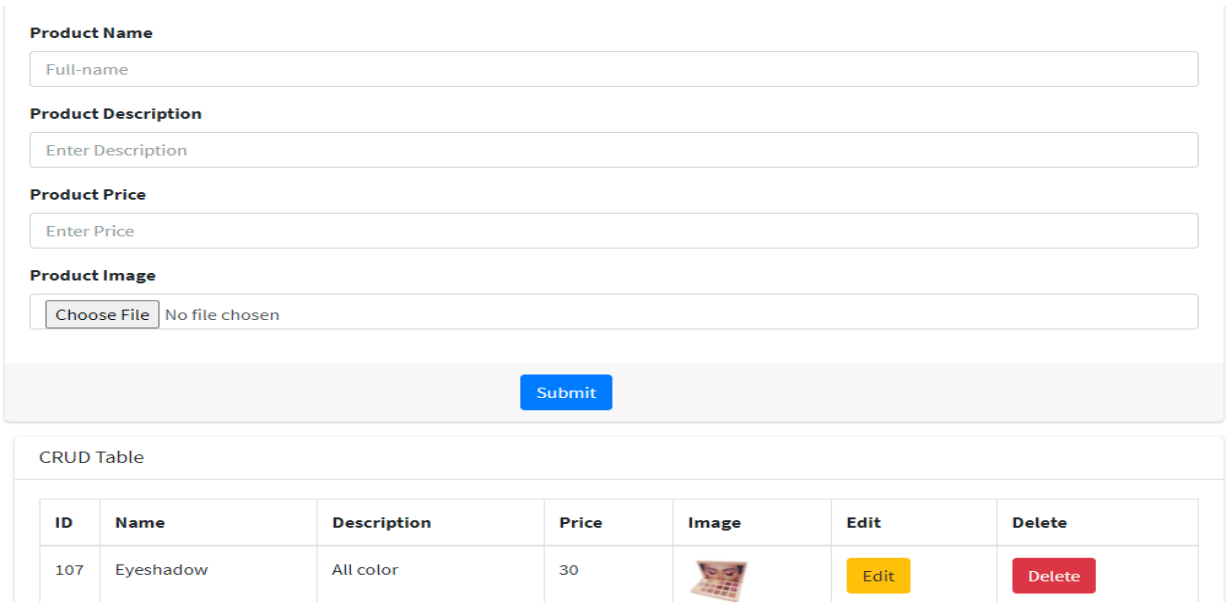
1-Vendor information:

In this page vendor can see his information and update it.



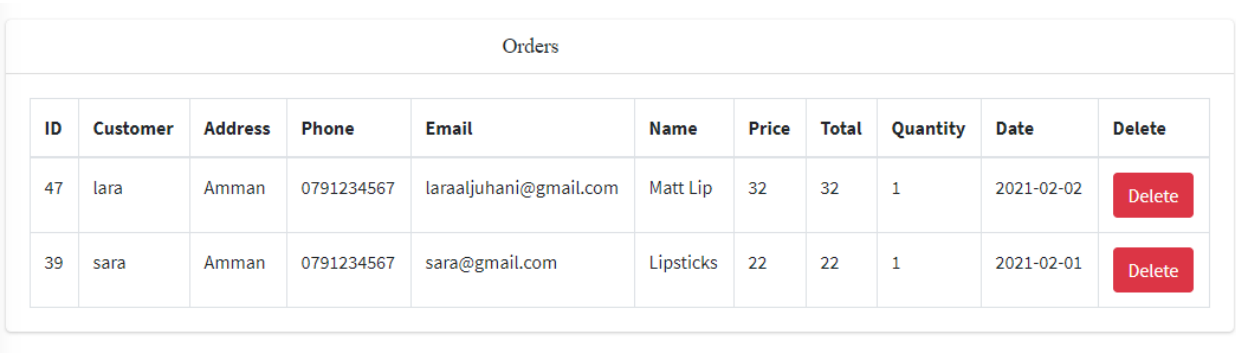
2-Vendor product:

In this vendor can create, edit and delete the product.



3-Vendor orders:

In this page vendor can see all order with the detail of customer and product.



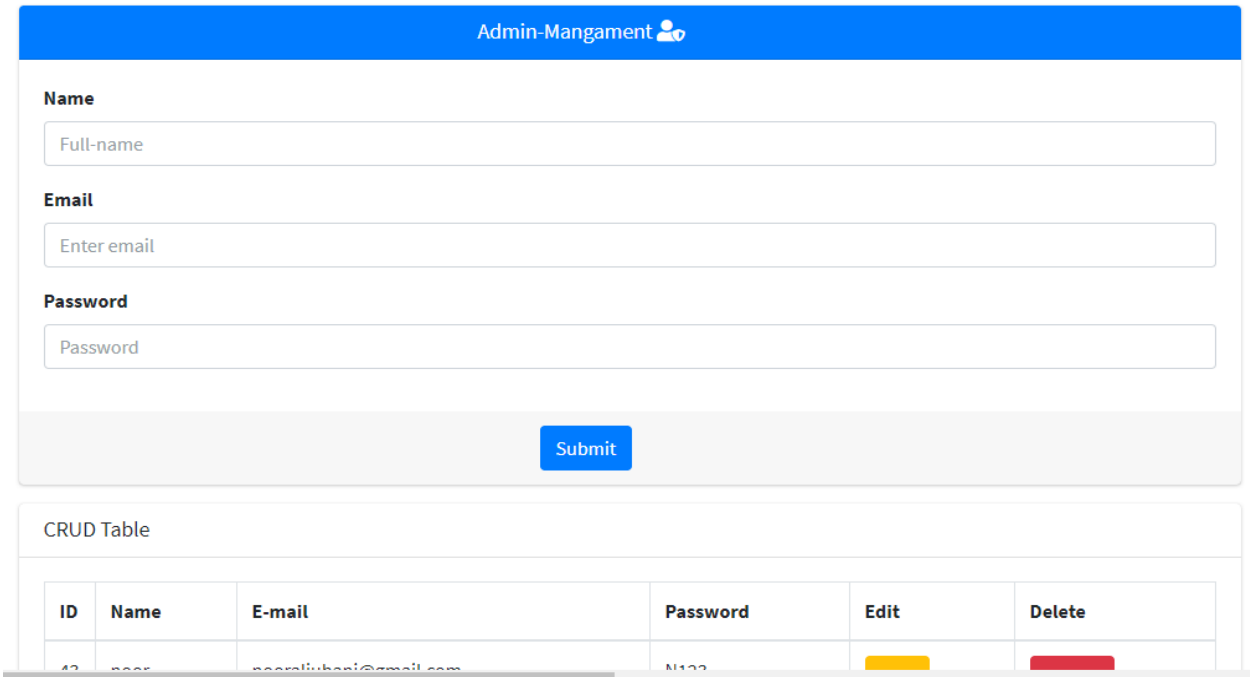
Orders

ID	Customer	Address	Phone	Email	Name	Price	Total	Quantity	Date	Delete
47	lara	Amman	0791234567	laraaljuhani@gmail.com	Matt Lip	32	32	1	2021-02-02	Delete
39	sara	Amman	0791234567	sara@gmail.com	Lipsticks	22	22	1	2021-02-01	Delete

-Admin dashboard

1-Admin page:

In this page admin can create, edit and delete admin.



Admin-Mangament

Name

Full-name

Email

Enter email

Password

Password

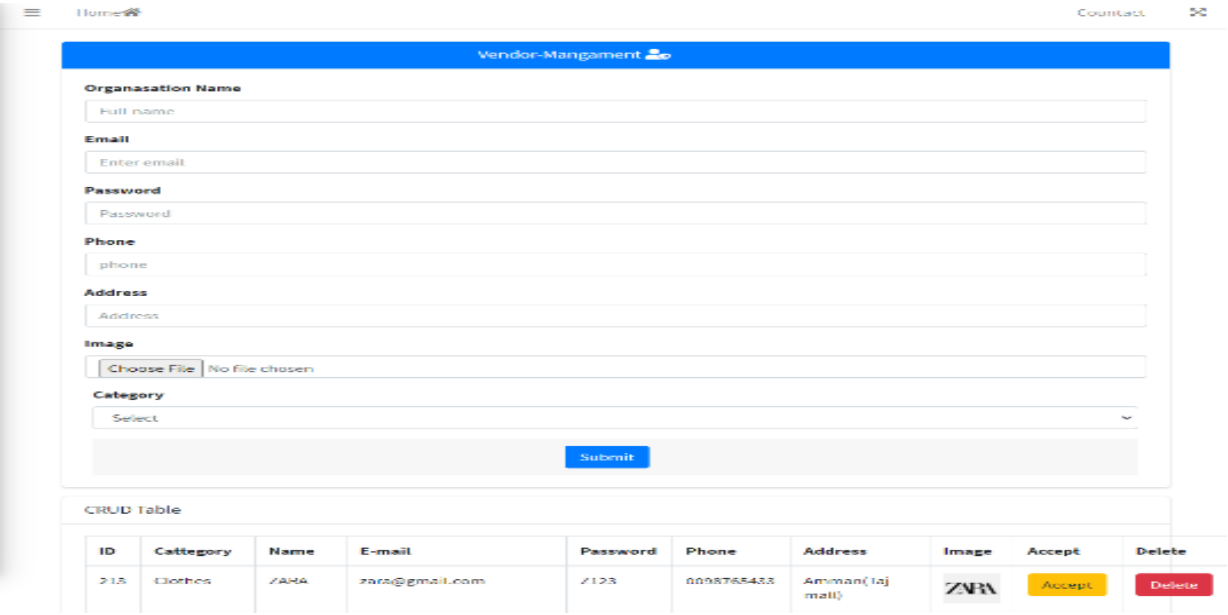
Submit

CRUD Table

ID	Name	E-mail	Password	Edit	Delete

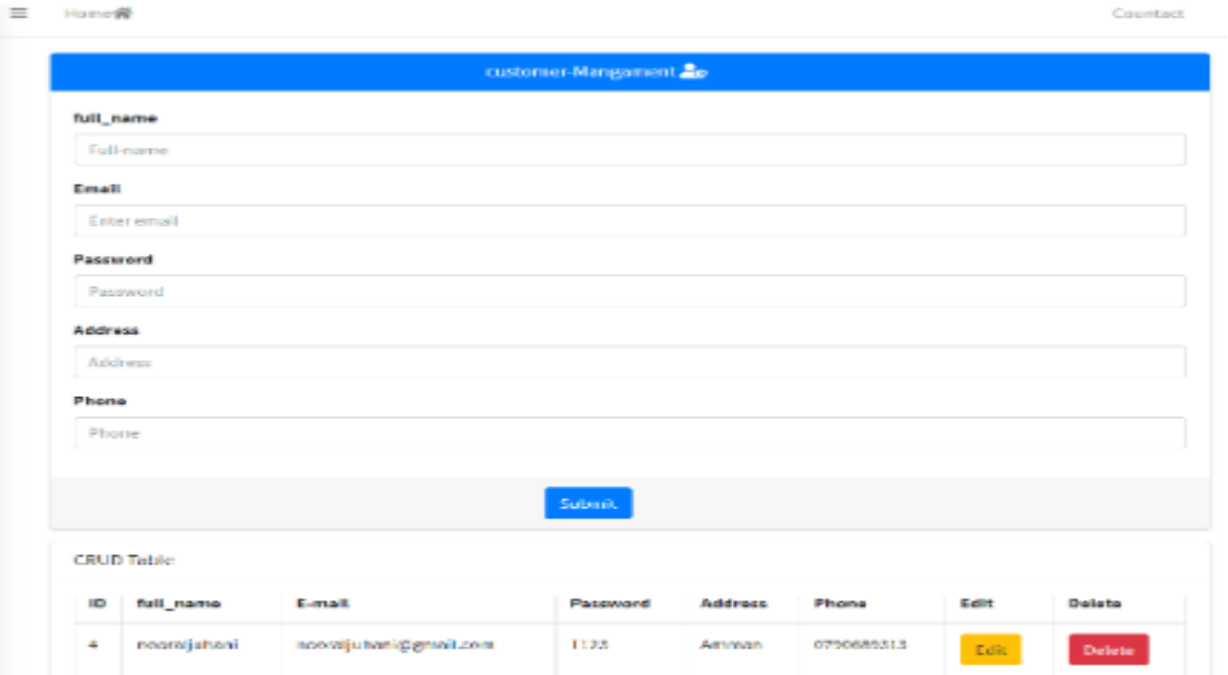
2-Vendor management:

In this page admin can create, accept and delete vendors.



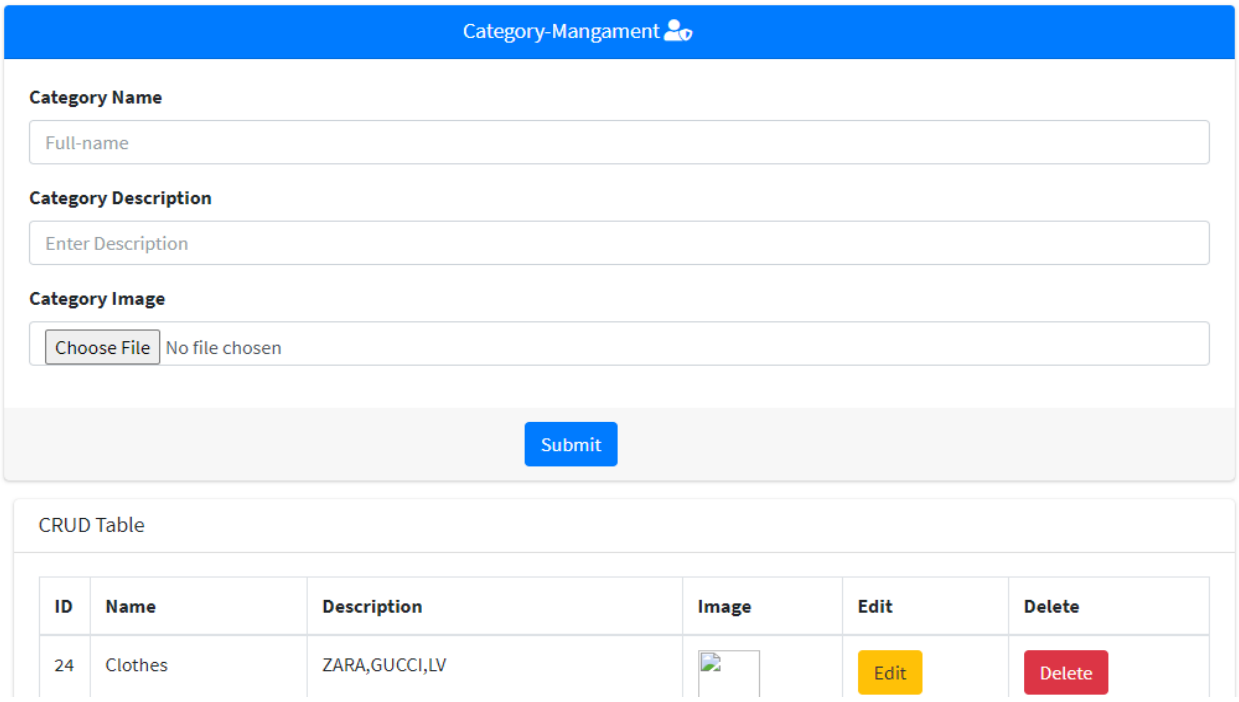
3-Customer management:

In this page admin can create, edit and delete customer.



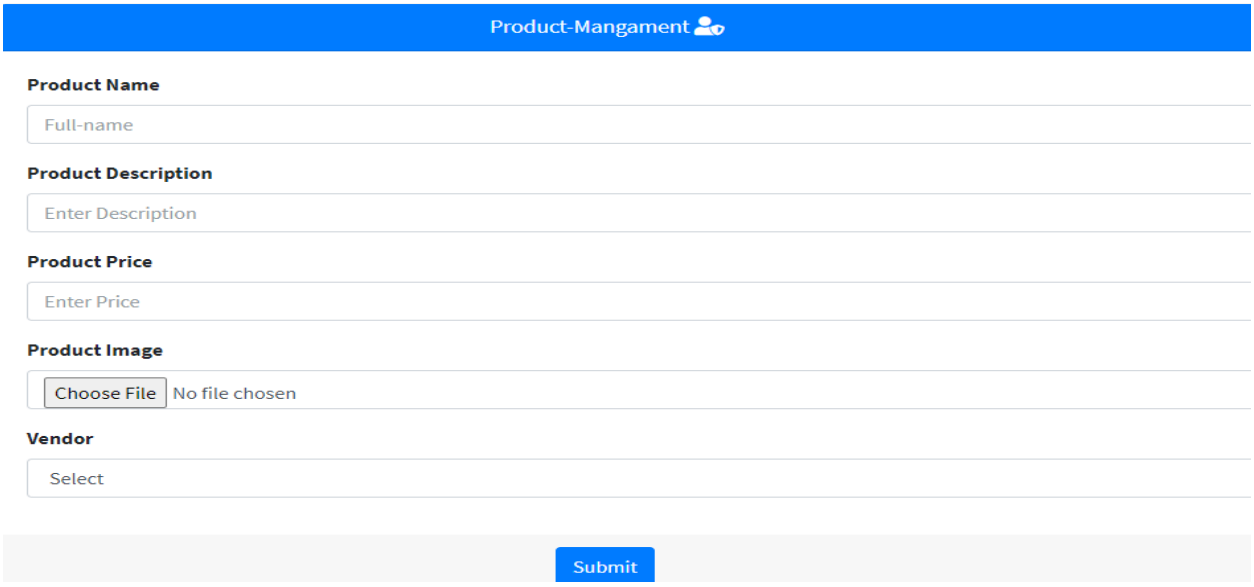
4-Category management:

In this page admin can create, edit and delete category.



5-Product management:

In this page admin can create, edit and delete product.



Summary

Shows all implementation details for project including the coding, database and graphical user interface implementation.

The output of this chapter is very necessary for the next chapter, whereby the testing and evaluation take place.

- Test cases

Number	Test cases
1	Login Proper validations of the Login page (e.g., email and password fields). The user logs in if the password entered correctly and fails to open their account if the pass is wrong.
2	Remove/add/update items list easily. Besides, price alterations along with vouchers, discounts should be automatically accounted into the total price.
3	Verify that the user is able to navigate through all the products across different categories.
4	Verify that the user can buy products added to the cart after signing in to the application (or as per the functionality of the website).
5	Verify that clicking submits button after entering all the required fields, submits the data to the server.

Conclusion

Any existing or potential vendor who is about to commence his commercial activities online needs lots of things to be considered. One of these things is to decide which open-source multi-vendor ecommerce platform to use in order to be successful and make the online business efficient. There are quite a few instruments that can be used for that but the experts recommend only the best ecommerce shopping mall solutions. So, the question is how to start multi-vendor ecommerce website, then? The answer is very simple, which is to stick with already made tool that will allow to kick start commercial activities immediately! This is called the Multi-Vendor.

Reference

-www.smartdraw.com

-www.lucidchart.com

-erdplus.com

-placeit.net

-www.ecwid.com

APPENDIX A

Source Code

```
<?php

  include('../includes/adminClass.php');
 $x = new admin();

    if(isset($_POST['submit'])){

    $x->email       = $_POST['email'];
    $x->password    = $_POST['password'];
    $x->name        = $_POST['name'];
    $q=$x->create();

    if($q){
        header("location:Admins_Management.php");

    }

}

   include("test.php");
 ?>
<!--body start-->
    <!-- Main content -->
                  <!-- form start -->
    <div  style="margin-left: 300px;margin-top: 20px;width: 2000px">
    <section class="content">
      <div class="container-fluid">
        <div class="row">
          <!-- left column -->
          <div class="col-md-6">
            <!-- general form elements -->
            <div class="card card-primary">
              <div class="card-header">
                <h3 class="card-title" style="margin-left: 350px">Admin-
Mangament  <i class="fas fa-user-shield"></i></h3>
              </div>
              <!-- /.card-header -->

              <form method="post">
                <div class="card-body">
                  <div class="form-group">
                    <label for="exampleInputPassword1">Name</label>
                    <input type="text" class="form-control" name="name"
placeholder="Full-name">
                  </div>
                  <div class="form-group">
                    <label for="exampleInputEmail1">Email</label>
```

```
                    <input type="email" name="email" class="form-control"
placeholder="Enter email">
                  </div>
                  <div class="form-group">
                    <label for="exampleInputPassword1">Password</label>
                    <input type="text" class="form-control" name="password"
placeholder="Password">
                  </div>

                </div>
                <!-- /.card-body -->

                <div class="card-footer">
                  <button type="submit" class="btn btn-primary" name="submit"
style="margin-left: 400px">Submit</button>
                </div>
              </form>
            </div>
            <!-- /.card -->
</div>
</div>
</div>
</section>
<!--end form-->
  <!-- table start -->
   <section class="content">
      <div class="container-fluid">
        <div class="row">
          <div class="col-md-6">
            <div class="card">
              <div class="card-header">
                <h3 class="card-title">CRUD Table</h3>
              </div>
              <!-- /.card-header -->
              <div class="card-body">
                <table class="table table-bordered">
                  <thead>
                    <tr>
                      <th style="width: 10px">ID</th>
                      <th>Name</th>
                      <th>E-mail</th>
                      <th>Password</th>
                       <th>Edit</th>
                        <th>Delete</th>

                    </tr>
                  </thead>
                  <tbody>
                    <?php
                          if($data=$x->readAll()){

                                        foreach ($data as $key => $value) {
                                            foreach ($value as $k => $v)
{$row[$k]=$v;}
                                                echo "<tr>";
```

```
                                                    echo
"<td>{$row['admin_id']}</td>";
                                                    echo
"<td>{$row['name']}</td>";
                                                    echo
"<td>{$row['email']}</td>";

                                                    echo
"<td>{$row['password']}</td>";
                                                    echo "<td><a
href='Edit_Admin.php?id={$row['admin_id']}' class='btn btn-
warning'>Edit</a></td>";
                                                    echo "<td><a
href='Delete_Admin.php?id={$row['admin_id']}' class='btn btn-
danger'>Delete</a></td>";
                                                    echo "</tr>";


                                            }


                                                }
                        ?>
                    </tbody>
                </table>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>

  <!-- table end -->


<!-- Body End -->

<?php

require('DBconnection.php');

class admin extends dbconnection{

        public $admin_id;
        public $email;
        public $password;
        public $name;




        public function create(){
                $query = "INSERT INTO admin(email,password,name)
                                VALUES('$this->email','$this-
>password','$this->name')";
                return $this->performQuery($query);
```

```
        }

        public function readAll(){
                $query  = "SELECT * FROM admin";
                $result = $this->performQuery($query);
                return $this->fetchAll($result);
        }
        public function readById($id){
                $query  = "SELECT * FROM admin WHERE admin_id = $id";
                $result = $this->performQuery($query);
                return $this->fetchAll($result);
        }
        public function update($id){
                $query = "UPDATE admin SET      email      = '$this->email',
                                                        password
        = '$this->password',
                                                                name
= '$this->name'
                                                        WHERE
admin_id   = $id";
                return $this->performQuery($query);
        }
        public function delete($id){
                $query = "DELETE FROM admin WHERE admin_id = $id";
                $this->performQuery($query);
        }
        public function login($email,$password){
                $query  = "SELECT * FROM admin
                           WHERE email = '$email' AND
                                        password   = '$password' ";

                $result = $this->performQuery($query);
                return $this->fetchAll($result);
        }
}
```