CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The main purpose of this project is to develop a web application that allowscustomers to order products through online. Furthermore, the project is built to allow the administrator(dealer) to control the product details. It's hectic for shopkeepers to place huge orders manually which requires manpower , thus the application allows shopkeepers to buy products via the internet.

The application has been completed and is divided into two parts:

- Admin side
- Shopkeeper side

Admin Side include creating a username and password, category and adding items to specific categories, editing items, querying sale data,querying database data, and logging out. The system allows the shopkeeper to sign upand log in. The shopkeepers may see all of the purchases made as well as the current availability of the products. Every order is recorded in a database. The dealer can viewthe received orders in the dashboard.

Create an Account, Sign-in, Select Products, Search Products, View Products, Buy Products, Continue Shopping, View Cart, Checkout, Confirm, and Logout are the various functionalities in the customer side

1.2 OBJECTIVE OF THE PROJECT

To construct an online wholesale web application to automate all the system's processes.

Types of maintenance,

- ➢ To maintain customer details
- ➢ To generate the report
- ➢ To improve customer satisfaction

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Shopkeeper must make a call or to visit the dealer and order products.This takes more time to process orders. Creating and maintaining reports takes a significant amount of time. If there is a need to get information about the customer, the entire report is re-typed. This has a significant impact on the system's authentication.

2.1.1 drawback of existing system

- The current system is not customer pleasant since data is not kept in a structuredand formatted manner
- The customer must visit a store and place an order.
- Description of the product obtained only on manually.
- Not in reach of distant users.
- Accuracy not guaranteed.

2.2 PROPOSED SYSTEM

The following actions are part of the development of our new system, which aims to automate the entire process while keeping the database integration strategy in mind. The programmer provides customer friendliness with numerous controls. The solution makes project management considerably easier. There is no danger of data manipulation.

2.2.1 Advantages of Proposed System

- Category will be listed to identify the product easily
- Technique for discovering reports that is simple and quick.
- It can be accessed over the Internet.
- The amount of manpower required is really low due to database report.
- Data may be kept for a longer period of time.

2.3 FEASIBILITY STUDY

A feasibility study is a detailed analysis that considers all of the critical aspects of a proposed project in order to determine the likelihood of it succeeding. Although feasibility studies can help project managers determine the risk and return of pursuing a plan of action, several steps should be considered before moving forward.

2.3.1 Economic Feasibility

By making the shop available in online will increase the ultimate income By converting from a manual to a digital system has several practical advantages. The user can receive solutions to inquiries, and the argument for any capital expenditure is that it would cut spending or enhance the quality of service or commodities, resulting in improved profits.

2.3.2 Operational Feasibility

The user can easily access the website and purchase the product .This system is user friendly new users can be able to access and purchase the product. Thereis no difficulty in placing the order. Once the customer create the account they can login id and there is no time delay to login into their profile. This systemcan accommodate the company's present inventory management processes.

2.3.3 Technical Feasibility

The present system is designed to address the shortcomings of the previous system. The system is simple to understand and use. There is no data loss and time delay in the present system.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

Processor	: I3 PROCESSOR
RAM	: 4GB RAM
Hard Disk	: 1 TB
Key Board	: LOGITECH
Monitor	: 14” color monitor
Mouse	: LOGITECH MOUSE

3.2 SOFTWARE REQUIREMENTS

Operating System	: Windows 10
Front End	: HTML/CSS, JavaScript
Backend	: MySQL (WampServer),PHP

3.2.1 FRONT END

HTML

HTML is a standardised system that creates the structure for just about every page that we find and use on the web. HTML works to build this structure by using tags that tell browsers what to do with text.

An HTML document is based on a file containing hypertext markup language. HTML tells, Web browsers, how to display text. There are three categories of HTML: transitional, strict, and frameset.

Features of HTML

- It is easy to learn and easy to use.
- It is platform-independent.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to the text.
- It is a markup language.

CSS

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media . It can control the layout of multiple web pages all at once .There are three types of CSS which are given below: Inline CSS. Internal or Embedded CSS. External CSS.

Features of CSS

- With CSS, you can control the color, font, the size of text, the spacing between elements
- Better User Experience. It also allows for user-friendly formatting.
- Quicker Development Time.
- Easy Formatting Changes.
- Compatibility Across Devices.

JAVASCRIPT:

JavaScript makes web pages dynamic. Before JavaScript, web pages were built only with HTML and CSS. HTML and CSS are only capable of creating static pages that can be

styled but not interactive aside from hyperlinks. Some of the most popular websites are built with JavaScript, including Google, YouTube, and Facebook.

Features of JAVASCRIPT

- Scripting Language.
- Interpreter Based.
- Event Handling.
- Light Weight.
- Case Sensitive

3.2.2 BACKEND

MySQL:

MySQL is a free-to-use relational database management system. A relational database organises data into one or more data tables, each of which includes related data types, allowing the data to be organised.

It works within the operating system to build a relational database in a computer's storage system. It also focus on managing users, providing network access, and simplifying database integrity testing and backup generation.

Features

- MySQL is a relational database management system
- MySQL is easy to use
- It is secure
- Client/ Server Architecture
- Speed
- High Flexibility

Features of Wamp-server

WAMP just requires the download and execution of one zip, tar, 7z, or exe file, with little or no configuration of the web server's numerous components. Microsoft Visual C++ 2017 Redistributable is required for the Windows version of WAMPP.

PHP:

PHP stands for Hypertext Pre-processor. It is a widely-used, free, and efficient server-side scripting language that can be incorporated in HTML and is designed for web-based technologies. It is a "Server-side HTML embedded scripting language" since it runs on the server. The types of scripting languages are client side & server side

Features of PHP:

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database

CHAPTER 4

PROJECT DESCRIPTION

4.1 OVERVIEW OF THE PROJECT

The aim of this project is to develop a web application that allows customer to order products through online. Furthermore, the project is built to allow the administrator(dealer) to control the product details. It's hectic for shopkeepers to place hugeorders manually which requires more manpower , thus the application allows customer to buy products via the internet.

4.2 MODULE DESCRIPTION

The project contains following modules

- My cart
- Admin
- User Login
- My Account
- Message
- Description
- User Registration

4.2.1 Customer Details

Customers are the important asset for any company. The details such as customer name, customer contact number, customer email id is collected. A separate table is designed to store information in database.

4.2.2 Item Details

Customers can purchase items from the comfort of their own homes or work places. Different kind of food products are delivered on customers comfort zone. Cost is also affordable and valuable for the products.

4.2.3 Message Details

If any defects in the food products or if the product is changed, the customer complaint at any time.

4.2.4 Report Details

Various user friendly reports are supported, they are product report, Purchase report, sales report and customer report.

4.2.5 Order Details

Order details module is designed such a way to collect all information pertaining to the products being ordered. For every ordered product the details like date of order, product name, quantity, and the customer id is maintained.

4.2.6 Help Services

In contact us option, mail id , phone no, & address are given for user references.

4.3 INPUT DESIGN:

The input design process involves converting input requirements into a machine-readableformat. The goal of input design is to develop an easy-to-follow, user-friendly input layout that avoids operator mistakes.

4.3.1 Registration Form:

The registration form gets the various information from the user and store it in the database table. The registration form is designed using php form to get details from the userthe details like user name ,email id and password will be get as input from the user for registration

4.3.2 Checkout Form:

The checkout form is the designed using php form it will get the details from the user toplace the order the various details getting from the users to place order is name, mail id, address, payment method and pin code

4.3.3 Contact Form:

The contact form in user side is designed using php form to get the queries from the user with gets the following details from the user like name, email id, mobile number and the usercan give the message in say something container

4.3.4 Adding products to the cart:

User can search for the product in shop menu and the can add the particular product to cartwhich they need to purchase and they can check the product in cart menu in the top right.

4.3.5 Adding products to the database:

In the admin side they can add new products by clicking product menu and the admin can add new product which will be displayed in the user side in the shop the following details need to be feeded by the admin to add the product they are product name, product price and upload the image by choosing the file from the system

4.4 OUTPUT DESIGN

The output designs are displayed some different report formats. Different output design willimprove the clarity and performing of output. The output designs are classified into individuals and group of tables is possible. And also display the reports are in lab tests, cross matching, and issue details is available in my project. It is used to check the collection of particular time of the period.

4.4.1 Product Display:

The product which will be added by the admin will be shown here and the price of the product is shown in the top left of the image and below the product image the quantity of theproduct is displayed in rectangular box and below that add to cart button is displayed by clicking it the product will be added to the cart

4.4.2 Orders Display:

The order placed by the user is displayed in the orders section. By clicking the orders menuwe can able to display the fields like date, name, phone number, address, payment method, orderlist & total price. And also we see the payment status.

4.4.3 User Accounts:

In the admin portal the registered user details are displayed in the container format Which display the fields like user name, email id and the user type

4.4.4 Messages:

In the admin portal the messages are displayed where the user gives message in the contactoption in the user portal which contains the fields like user name, mobile number and the messages will be displayed

4.4.5 Reports:

Admin can get the overall sales report for the total sales by giving the start date and the enddate in report section and by clicking get report the admin can get the sales report of overall sales

4.5 DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database managementsystem manages the data accordingly.

4.6 DATAFLOW DIAGRAM

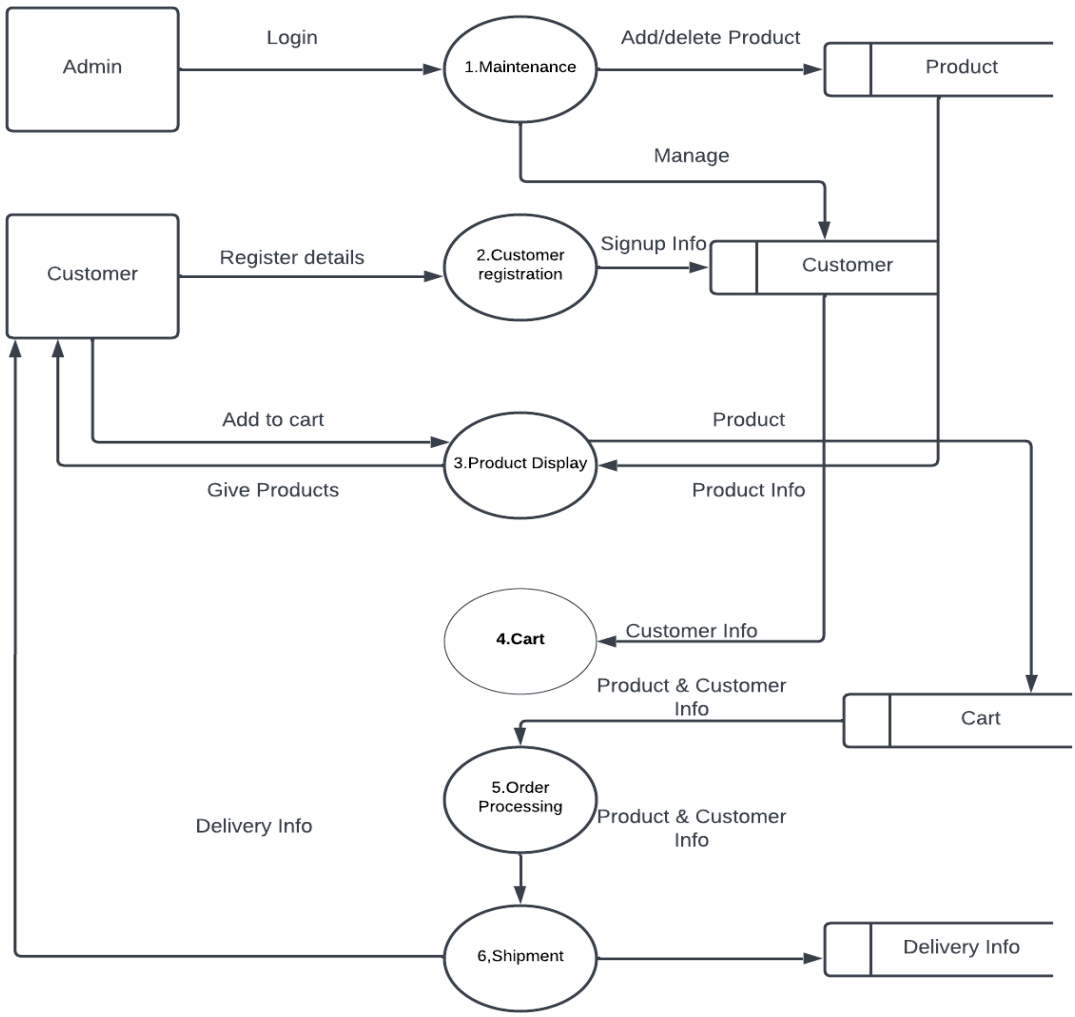

Figure 4.1 Dataflow Diagram

4.6.1 DFD Level Admin

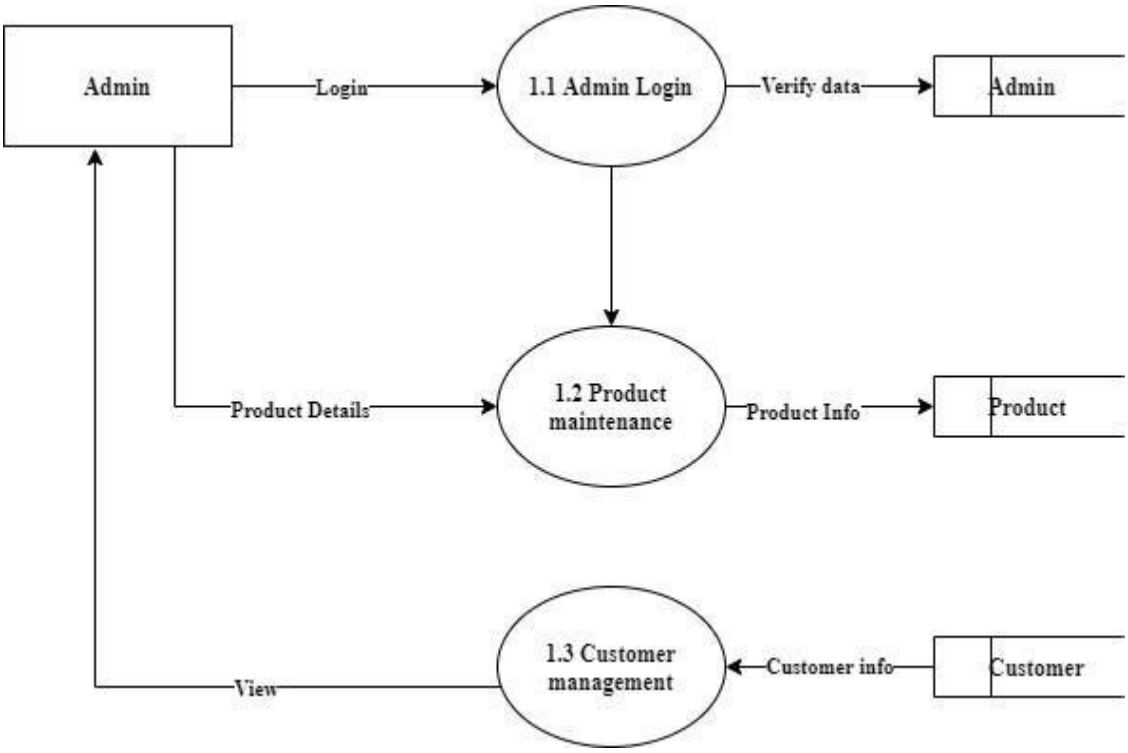

Figure 4.2 DFD Level Admin

4.6.2 DFD Level Customer Registration

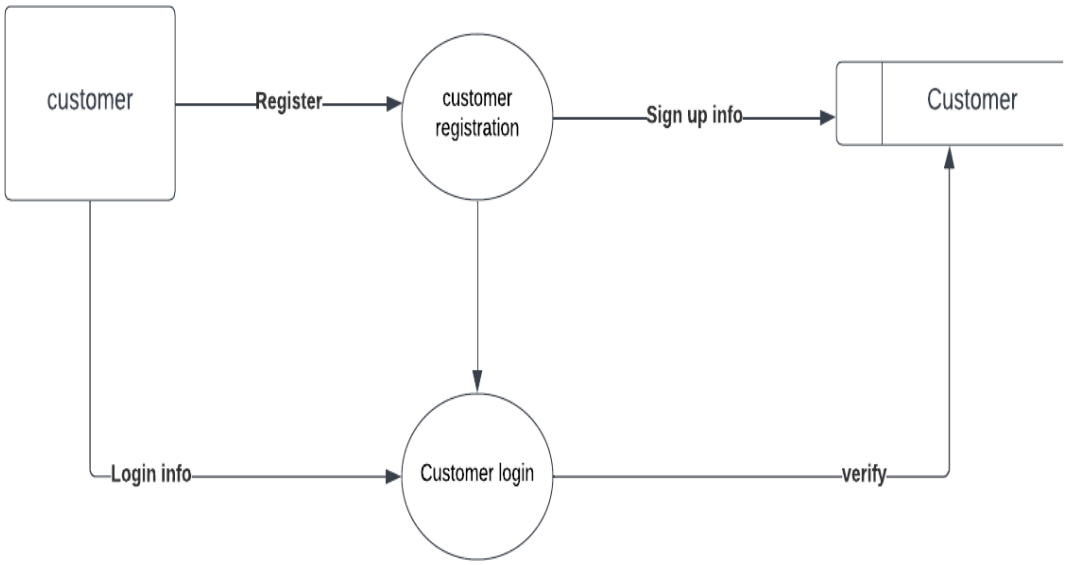

Figure 4.3 DFD Level for Customer Registration

4.7 ENTITY RELATIONSHIP DIAGRAM

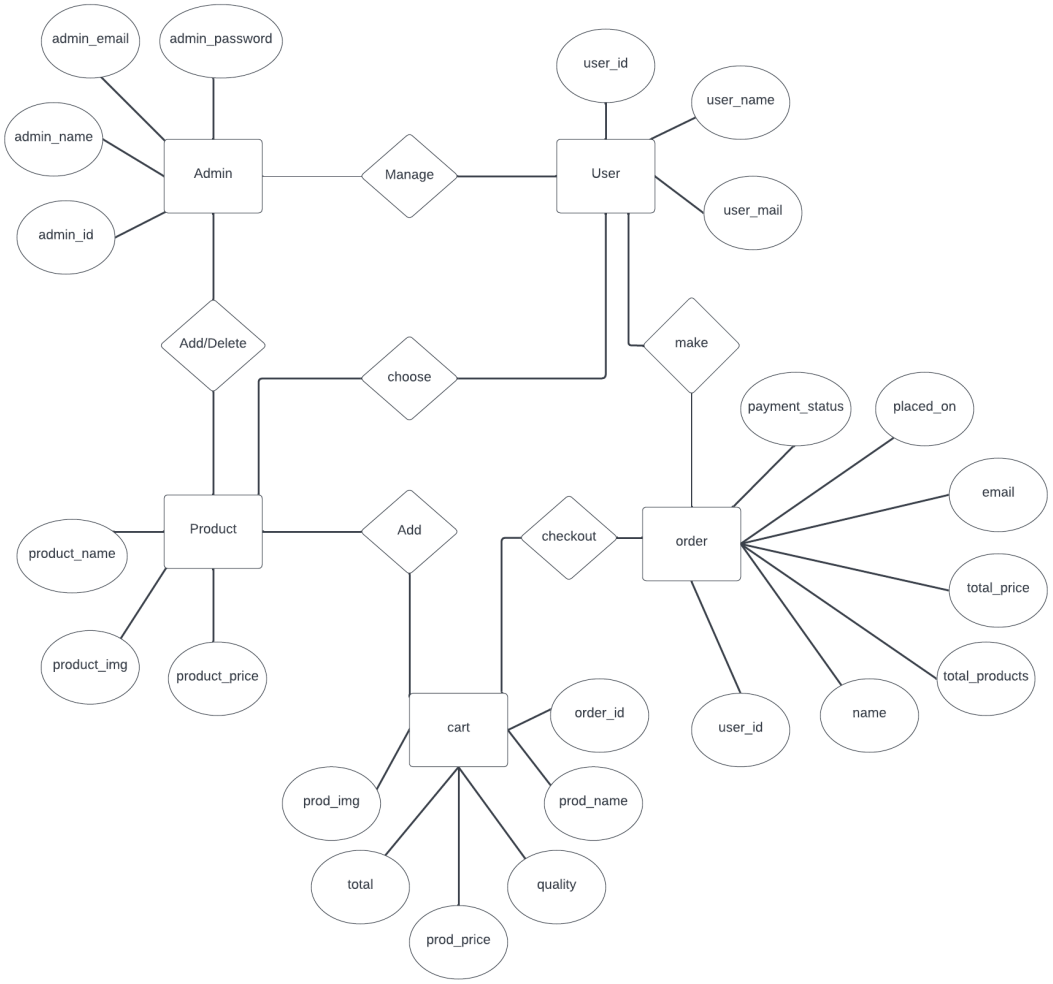

Figure 4.4 Entity Relationship Diagram

4.8 DATABASE TABLE DESIGN

In php MyAdmin, MySQL database is created to store the user data for the following tables.

- Order
- Product
- Cart
- User
- Message

Table 4.8.1 Order

SNO	FIELD NAME	DATA TYPE	SIZE
.1.	Order_Id	Int	10
2.	User_name	Varchar	20
3.	Address	Varchar	20
4.	Mobile number	Big int	10
5.	Email	Varchar	20
6.	Payment_method	Varchar	20
7.	total_price	Varchar	50
8.	Placed_on	Varchar	20
9.	Total products	Int	10
10.	Payment_status	Varchar	30

Table 4.8.2 Product

SNO	FIELD NAME	DATA TYPE	SIZE
1.	Order_Id	Int	10
2.	Product_name	Varchar	20
3.	Product amt	Int	10
4.	Product total amt	Int	10
5.	Product img		
6.	Product size	Varchar	20

Table 4.8.3 Cart

SNO	FIELD NAME	DATA TYPE	SIZE
1.	Id	Int	10
2.	Product qty	Int	11
3.	Price	Int	10
4.	Name	Varchar	100

Table 4.8.4 User

SNO	FIELD NAME	DATA TYPE	SIZE
1.	User_id	int	10
2.	Email	varchar	50
3.	Password	varchar	50
4.	User_type	varchar	50
5.	Name	varchar	50

Table 4.8.5 Message

SNO	FIELD NAME	DATA TYPE	SIZE
1.	Order_id	Int	10
2.	User id	Int	20
3.	Name	Varchar	10
4.	Email	Varchar	20
5.	Number	Varchar	12
7.	Message	Varchar	500

CHAPTER 5

SYSTEM TESTING

System testing assumes that if all aspects of the system are accurate, system testing will be useful as a user-oriented vehicle prior to implementation. If a software does not suit the demands of its users, it is useless. System testing detects faults, presents a suggestion to the administrator, modifies the alteration, and verifies the output's dependability. Before going live, the system is checked to see if the necessary software andhardware are in place to complete the project. To guarantee that this project is correct, it haspassed the following testing methods.

- Unit testing
- Integration testing
- Validation testing

5.1 UNIT TESTING

Unit testing's primary purpose is to isolate the smallest piece of tested software in the application from the rest of the code and evaluate whether it functions precisely as you anticipate. Each item is tested independently before being integrated into modules, and the interfaces are checked first. First, procedure level testing is performed. The mistakes that occurred are identified and eliminated by providing incorrect inputs. The formlevel testing is then performed. For example, data storage to the table is done correctly. Eachform is treated as an independent entity and checked for faults in this method. Every user input is checked for a valid range of values.

Test Case 1

Module : Admin Login

Test Type : Loading the form for administrator

Input : Email and Password

Expected Output : Display admin dashboard detail

Sample Test

Input : abcd@gmail.com & 1234

Output :Redirect to Admin Page and display the admin dashboard details

Analysis :In this form, the email and password has been tested by correct format.If the email and password is mismatched to data in the database, it will display as incorrect email or password at the top.

Result : The expected output is same as original output

Test Case 2

Module : Customer Login

Test Type : Loading the page for customer

Input : Email and Password

Expected Output : It will display the home page & Latest products

Sample Test

Input : abcd@gmail.com & 1234

Output :Redirect to Customer Page and display the products

Analysis :In this form, the email and password has been tested by correct format.If the email and password is mismatched to data in the database, it will display as incorrect email or password at the top.

Result : The expected output is same as original output

Test Case 3

Module : Customer Login

Test Type : Checking the username and password
Input : Email and Password
Expected Output : incorrect the email are password

Sample Test

Input : abcde@gmail.com & 12345

Output : It shows the customer incorrect user name and password

Analysis :In this form, the email and password has been tested.If the email and password is mismatched to data in the database, it will display as incorrect email or password at the top.
Result : The expected output is same as original output

5.2 INTEGRATION TESTING

Integration testing is also known as integration and Individual software modules are assembled and evaluated as a group in this step of software testing, abbreviated "I&T." It happens between unit testing and system testing. Integration testing takes unit-tested modules as input, collects them into bigger aggregates, performs tests described in an integration test plan to those aggregates, and produces an integrated system ready for systemtesting as an output.

The goal of integration testing is to ensure that primary design elements meet functional and reliability criteria. These "design objects," i.e., assemblages (or groupings ofunits), are put through their paces via their interfaces utilizing Black box testing, with successand failure scenarios simulated using suitable parameter and data inputs.

Each module is thoroughly tested. After all of the modules have been tested, the complete system is tested using test data that has been particularly prepared to

demonstrate that the system will work successfully in all of its features. As a result, system testing serves as botha validation that everything is in order and a chance to demonstrate to the user that the systemworks.

Test Case 1

Module: Admin Menus
Test Type: Add product & product name & product price
Input: Product details is updated
Output: Product details is completed

Sample Test

Input: On Clicking Login, Select the product.
Output: Respective product will be in the cart.
Analysis: Product will be displayed in the cart

5.3VALIDATION TESTING

Validation may be described in a variety of ways, but one basic definition is that it is statedin the software requirement specification, a document that covers all user-visible attributes of the product.

Test case 1

Module: User Register
Test Type: Register as new user
Input: Input to all fields Expected
Output : No Required field should not be empty

Sample Test

Input: “Ram, abcd@gmail.com, 234, 234, user”
Output: New user is successfully registered
Analysis: The expected output is same as the actual output

Test case 2

Module: Admin Register

Test Type: Register as new admin

Input: Input to all fields Expected

Output: No Required field should not be empty

Sample Test

Input: “Ram, abcd@gmail.com, 234, 234, admin”

Output: New admin is successfully registered

Analysis: The expected output is same as the actual output

Test case 3

Module: User Register

Test Type: validate the input type

Input: Input to all fields Expected

Output: No Required field should not be empty

Sample Test

Input: “Ram, abcdgmail.com, 234, 234, user”

Output : Required ‘@’ for gmail

Analysis: The expected output is same as the actual output

CHAPTER 6

SYSTEM IMPLEMENTATION

Windows, MySQL, and PHP (WAMP) are acronyms for Windows, Apache, MySQL, and PHP. It's a popular Windows replacement for XAMPP. XAMPP can be used to setup the running environment and needs little effort only. However WAMP is designed exclusively for Windows, developers may opt to use it. XAMPP, on the other hand, may be used on both Linux and Mac OS. Both platforms, however, are equally stable and offer the same functionality.

Step 1: Download the WAMP Server

The WampServer is downloaded from the website https://www.wampserver.com/en/. WampServer is available in two versions: 64-bits and 32-bits. The 64-bits version is used for this project.

Step 2: Start WAMP Server Install Process

Immediately after clicking on the downloaded file, the user will be prompted to select the desired language and click the OK button. The next screen shows the license agreement. Select the ‘I Agree’ radio button and then select the ‘Next’ button to continue the installation. The components that need to be present in the computer system for WampServer to function properly are listed on the next screen.

Step 3: Select Destination to Install WAMP Server

To install WAMP on the computer, user must first choose a location. User can, however, accept the default location and proceed. Press the 'Next' button to continue the installation.

Step 4: Select Start Menu Folder to Install WAMP

The user must choose the folder in which the program's shortcut should be created. User has the option of selecting any folder or continuing with the default option. This folder is easily accessible from the start. To continue, select the folder and click 'Next.'

Step 5: Ready to Install WAMP

The user will be presented with the Ready to Install Wizard ready to start the installation. To start the installation, just click the "Install" button. The WampServer is now extracting files to the place you specified. Allow the procedure to finish. When the status bar is entirely green, an information screen similar to the one below will appear. To proceed, click 'Next.'

Step 6: WAMP Server Installation Complete

The setup is complete; click 'Finish' to complete it.

Testing :

Once the coding process begins, the testing process has to be carried to make the correct execution of the coding. User can directly open the website by clicking the website link. Website opens in laptop, mobiles and Desktop.

Documentation:

The user guide is a product of the installation process and includes instructions on how to use the system and its flow.

Training and Support:

As the project is a web based application, the UI is simple to use. The application has in built training system in a video based manner so that the user is using the application to their fullest satisfaction.

CHAPTER 7

CONCLUSION AND FUTURE EHANCEMENT

7.1 CONCLUSION

This project **"WEBSITE FOR AGRO FOOD SALES"** has been successfully, computerand has been found to replace the existing system effectively. It is also possible to eliminate the human errors likely to creep into this kind of work in which bulk quantity ofdata has been processed. This project has been designed to suit all the exact needs. The Project helps admin to maintain all orders and details. It reduces the manual work.

All the objectives of this project are satisfied. The intermediate reports can be used for verification, if necessary in future. The system has been tested with sample data, with original data and the system is found to run well. The concern in which the proposed system will be implemented will find it more efficient. The functioning of the system canbe further enhanced in a number of ways, though an attempt has been made for security and high reliability.

This project is easy to implement and operate. All the features, which are given in this system, were successfully implemented and hence each and every module was tested andfound correct.

7.2 FUTURE ENHACEMENT

Enhancement refers to adding, modifying or redeveloping the code to support changes inthe specification. It is necessary to keep up with changing user requirements and the operational environment.

Following are the some the features which can be considered for future enhancements.

- Automatic Backup and restore the database
- Online transactions
- Inventory management

SAMPLE CODING

```
<?php

include 'config.php';
session_start();

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

   $email = mysqli_real_escape_string($conn, $_POST['email']);
   $pass = mysqli_real_escape_string($conn, md5($_POST['password']));

   $select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE email = '$email' AND
password = '$pass'") or die('query failed');

   if(mysqli_num_rows($select_users) > 0){

      $row = mysqli_fetch_assoc($select_users);

      if($row['user_type'] == 'admin'){

         $_SESSION['admin_name'] = $row['name'];
         $_SESSION['admin_email'] = $row['email'];
         $_SESSION['admin_id'] = $row['id'];
         header('location:admin_page.php');

      }elseif($row['user_type'] == 'user'){

         $_SESSION['user_name'] = $row['name'];
         $_SESSION['user_email'] = $row['email'];
         $_SESSION['user_id'] = $row['id'];
         header('location:home.php');

      }

   }else{
      $message[] = 'incorrect email or password!';
   }

}

?>

<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>login</title>

  <!-- font awesome cdn link  -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">

  <!-- custom css file link  -->
  <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php
if(isset($message)){
  foreach($message as $message){
    echo '
    <div class="message">
      <span>'.$message.'</span>
      <i class="fas fa-times" onclick="this.parentElement.remove();"></i>
    </div>
    ';
  }
}
?>

<div class="form-container">

  <form action="" method="post">
    <h3>login now</h3>
    <input type="email" name="email" placeholder="enter your email" required class="box">
    <input type="password" name="password" placeholder="enter your password" required
class="box">
    <input type="submit" name="submit" value="login now" class="btn">
    <p>don't have an account? <a href="register.php">register now</a></p>
  </form>

</div>

</body>
</html>
```

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if(!isset($user_id)){
   header('location:login.php');
}

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

   $product_name = $_POST['product_name'];
   $product_price = $_POST['product_price'];
   $product_image = $_POST['product_image'];
   $product_quantity = $_POST['product_quantity'];

   $check_cart_numbers = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
'$product_name' AND user_id = '$user_id'") or die('query failed');

   if(mysqli_num_rows($check_cart_numbers) > 0){
      $message[] = 'already added to cart!';
   }else{
      mysqli_query($conn, "INSERT INTO `cart`(user_id, name, price, quantity, image)
VALUES('$user_id', '$product_name', '$product_price', '$product_quantity', '$product_image')") or
die('query failed');
      $message[] = 'product added to cart!';
   }

}

?>

<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Home</title>

   <!-- font awesome cdn link  -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
```

```
   <!-- custom css file link  -->
   <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'header.php'; ?>

<section class="home">

   <div class="content">
      <h3>FARM FRESH</h3>
      <p>PROMOTING THE NATURAL PRODUCTS !!!</p>
      <a href="about.php" class="white-btn">discover more</a>
   </div>

</section>

<section class="products">

   <h1 class="title">latest products</h1>

   <div class="box-container">

      <?php
         $select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('query
failed');
         if(mysqli_num_rows($select_products) > 0){
            while($fetch_products = mysqli_fetch_assoc($select_products)){
      ?>
      <form action="" method="post" class="box">
      <img class="image" src="uploaded_img/<?php echo $fetch_products['image']; ?>" alt="">
      <div class="name"><?php echo $fetch_products['name']; ?></div>
      <div class="price">₹<?php echo $fetch_products['price']; ?>/-</div>
      <input type="number" min="1" name="product_quantity" value="1" class="qty">
      <input type="hidden" name="product_name" value="<?php echo $fetch_products['name'];
?>">
      <input type="hidden" name="product_price" value="<?php echo $fetch_products['price']; ?>">
      <input type="hidden" name="product_image" value="<?php echo $fetch_products['image'];
?>">
      <input type="submit" value="add to cart" name="add_to_cart" class="btn">
      </form>
      <?php
         }
      }else{
         echo '<p class="empty">no products added yet!</p>';
```

```
    }
    ?>
  </div>

  <div class="load-more" style="margin-top: 2rem; text-align:center">
    <a href="shop.php" class="option-btn">load more</a>
  </div>

</section>

<section class="about">

  <div class="flex">

    <div class="image">
      <img src="https://tradbull.com/wp-content/uploads/2021/01/TRADBULL-variety-of-beans.jpg" alt="">
    </div>

    <div class="content">
      <h3>about us</h3>
      <p>WE ARE SELLING THE BEST AGRICULTURAL FOOD PRODUCTS AND PROMOTING THE NATURAL FOOD PRODUCTS</p>
      <a href="about.php" class="btn">read more</a>
    </div>

  </div>

</section>

<section class="home-contact">

  <div class="content">
    <h3></h3>
    <p></p>
    <a href="contact.php" class="white-btn">contact us</a>
  </div>

</section>
<?php include 'footer.php'; ?>

<!-- custom js file link  -->
<script src="js/script.js"></script>

</body>
</html>
```

```
<?php

include 'config.php';

session_start();

$admin_id = $_SESSION['admin_id'];

if(!isset($admin_id)){
   header('location:login.php');
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>admin panel</title>

   <!-- font awesome cdn link  -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">

   <!-- custom admin css file link  -->
   <link rel="stylesheet" href="css/admin_style.css">

</head>
<body>

<?php include 'admin_header.php'; ?>

<!-- admin dashboard section starts  -->

<section class="dashboard">

   <h1 class="title">dashboard</h1>

   <div class="box-container">

      <div class="box">
         <?php
            $total_pendings = 0;
            $select_pending = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE
payment_status = 'pending'") or die('query failed');
```

```
        if(mysqli_num_rows($select_pending) > 0){
          while($fetch_pendings = mysqli_fetch_assoc($select_pending)){
            $total_price = $fetch_pendings['total_price'];
            $total_pendings += $total_price;
          };
        };
      ?>
      <h3>₹<?php echo $total_pendings; ?>/-</h3>
      <p>total pendings</p>
    </div>

    <div class="box">
      <?php
        $total_completed = 0;
        $select_completed = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE payment_status = 'completed'") or die('query failed');
        if(mysqli_num_rows($select_completed) > 0){
          while($fetch_completed = mysqli_fetch_assoc($select_completed)){
            $total_price = $fetch_completed['total_price'];
            $total_completed += $total_price;
          };
        };
      ?>
      <h3>₹<?php echo $total_completed; ?>/-</h3>
      <p>completed payments</p>
    </div>

    <div class="box">
      <?php
        $select_orders = mysqli_query($conn, "SELECT * FROM `orders`") or die('query failed');
        $number_of_orders = mysqli_num_rows($select_orders);
      ?>
      <h3><?php echo $number_of_orders; ?></h3>
      <p>order placed</p>
    </div>

    <div class="box">
      <?php
        $select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('query failed');
        $number_of_products = mysqli_num_rows($select_products);
      ?>
      <h3><?php echo $number_of_products; ?></h3>
      <p>products added</p>
    </div>

    <div class="box">
```

```
        <?php
          $select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE user_type =
'user'") or die('query failed');
          $number_of_users = mysqli_num_rows($select_users);
        ?>
        <h3><?php echo $number_of_users; ?></h3>
        <p>normal users</p>
      </div>

      <div class="box">
        <?php
          $select_admins = mysqli_query($conn, "SELECT * FROM `users` WHERE user_type =
'admin'") or die('query failed');
          $number_of_admins = mysqli_num_rows($select_admins);
        ?>
        <h3><?php echo $number_of_admins; ?></h3>
        <p>admin users</p>
      </div>

      <div class="box">
        <?php
          $select_account = mysqli_query($conn, "SELECT * FROM `users`") or die('query failed');
          $number_of_account = mysqli_num_rows($select_account);
        ?>
        <h3><?php echo $number_of_account; ?></h3>
        <p>total accounts</p>
      </div>

      <div class="box">
        <?php
          $select_messages = mysqli_query($conn, "SELECT * FROM `message`") or die('query
failed');
          $number_of_messages = mysqli_num_rows($select_messages);
        ?>
        <h3><?php echo $number_of_messages; ?></h3>
        <p>new messages</p>
      </div>

    </div>

</section>

<!-- admin dashboard section ends -->
```

```
<!-- custom admin js file link  -->
<script src="js/admin_script.js"></script>

</body>
</html>



<!--<?php

session_start();
include 'config.php';


$user_id = $_SESSION['user_id'];
$user_name=$_SESSION['user_name'] ;
$user_email=$_SESSION['user_email'];

if(!isset($user_id)){
   header('location:login.php');
}

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

   $name = mysqli_real_escape_string($conn, $_POST['name']);
   $number = $_POST['number'];
   $email = mysqli_real_escape_string($conn, $_POST['email']);
   $method = mysqli_real_escape_string($conn, $_POST['method']);
   $address = mysqli_real_escape_string($conn, 'flat no. '. $_POST['flat'].', '. $_POST['street'].', '.
$_POST['city'].', '. $_POST['country'].' - '. $_POST['pin_code']);
   $placed_on = date('Y-m-d');

   $cart_total = 0;
   $cart_products[] = '';

   $cart_query = mysqli_query($conn, "SELECT * FROM `cart` WHERE user_id = '$user_id'") or
die('query failed');
   if(mysqli_num_rows($cart_query) > 0){
      while($cart_item = mysqli_fetch_assoc($cart_query)){
         $cart_products[] = $cart_item['name'].' ('.$cart_item['quantity'].') ';
         $sub_total = ($cart_item['price'] * $cart_item['quantity']);
         $cart_total += $sub_total;
      }
   }

   $total_products = implode(', ',$cart_products);
```

```
   $order_query = mysqli_query($conn, "SELECT * FROM `orders` WHERE name = '$name' AND number = '$number' AND email = '$email' AND method = '$method' AND address = '$address' AND total_products = '$total_products' AND total_price = '$cart_total'") or die('query failed');

   if($cart_total == 0){
      $message[] = 'your cart is empty';
   }else{
      if(mysqli_num_rows($order_query) > 0){
         $message[] = 'order already placed!';
      }else{
         mysqli_query($conn, "INSERT INTO `orders`(user_id, name, number, email, method, address, total_products, total_price, placed_on) VALUES('$user_id', '$name', '$number', '$email', '$method', '$address', '$total_products', '$cart_total', '$placed_on')") or die('query failed');
         $message[] = 'order placed successfully!';
         mysqli_query($conn, "DELETE FROM `cart` WHERE user_id = '$user_id'") or die('query failed');
      }
   }

}

?>-->

<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>checkout</title>

   <!-- font awesome cdn link  -->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

   <!-- custom css file link  -->
   <link rel="stylesheet" href="css/style.css">

</head>
<body>

<?php include 'header.php'; ?>

<div class="heading">
   <h3>checkout</h3>
   <p> <a href="home.php">home</a> / checkout </p>
```

```
</div>

<section class="display-order">

  <?php
    $grand_total = 0;
    $select_cart = mysqli_query($conn, "SELECT * FROM `cart` WHERE user_id = '$user_id'") or die('query failed');
    if(mysqli_num_rows($select_cart) > 0){
      while($fetch_cart = mysqli_fetch_assoc($select_cart)){
        $total_price = ($fetch_cart['price'] * $fetch_cart['quantity']);
        $grand_total += $total_price;
  ?>
  <p> <?php echo $fetch_cart['name']; ?> <span>(<?php echo '₹'.$fetch_cart['price'].'/-' x '. $fetch_cart['quantity']; ?>)</span> </p>
  <?php
    }
  }else{
    echo '<p class="empty">your cart is empty</p>';
  }
  ?>
  <div class="grand-total"> grand total : <span>₹<?php echo $grand_total; ?>/-</span> </div>

</section>

<section class="checkout">

  <form action="" method="post">
    <h3>place your order</h3>
    <div class="flex">
      <div class="inputBox">
        <span>your name :</span>
        <input type="text" name="name" required placeholder="enter your name" value="<?php echo $user_name; ?>">
      </div>
      <div class="inputBox">
        <span>your number :</span>
        <input type="number" name="number" required placeholder="enter your number">
      </div>
      <div class="inputBox">
        <span>your email :</span>
        <input type="email" name="email" required placeholder="enter your email" value="<?php echo $user_email; ?>">
      </div>
      <div class="inputBox">
        <span>payment method :</span>
        <select name="method">
```

```
            <option value="cash on delivery">cash on delivery</option>
          </select>
        </div>
        <div class="inputBox">
          <span>Door no:</span>
          <input type="number" min="0" name="flat" required placeholder="e.g. flat no.">
        </div>
        <div class="inputBox">
          <span>address  :</span>
          <input type="text" name="street" required placeholder="e.g. street name">
        </div>
        <div class="inputBox">
          <span>city :</span>
          <input type="text" name="city" required placeholder="e.g. mumbai">
        </div>
        <div class="inputBox">
          <span>state :</span>
          <input type="text" name="state" required placeholder="e.g. maharashtra">
        </div>
        <div class="inputBox">
          <span>country :</span>
          <input type="text" name="country" required placeholder="e.g. india">
        </div>
        <div class="inputBox">
          <span>pin code :</span>
          <input type="number" min="0" name="pin_code" required placeholder="e.g. 123456">
        </div>
      </div>
      <input type="submit" value="order now" class="btn" name="order_btn">
    </form>

</section>
-->

<?php include 'footer.php'; ?>

<!-- custom js file link  -->
<script src="js/script.js"></script>

</body>
</html>
```

SAMPLE IMAGES

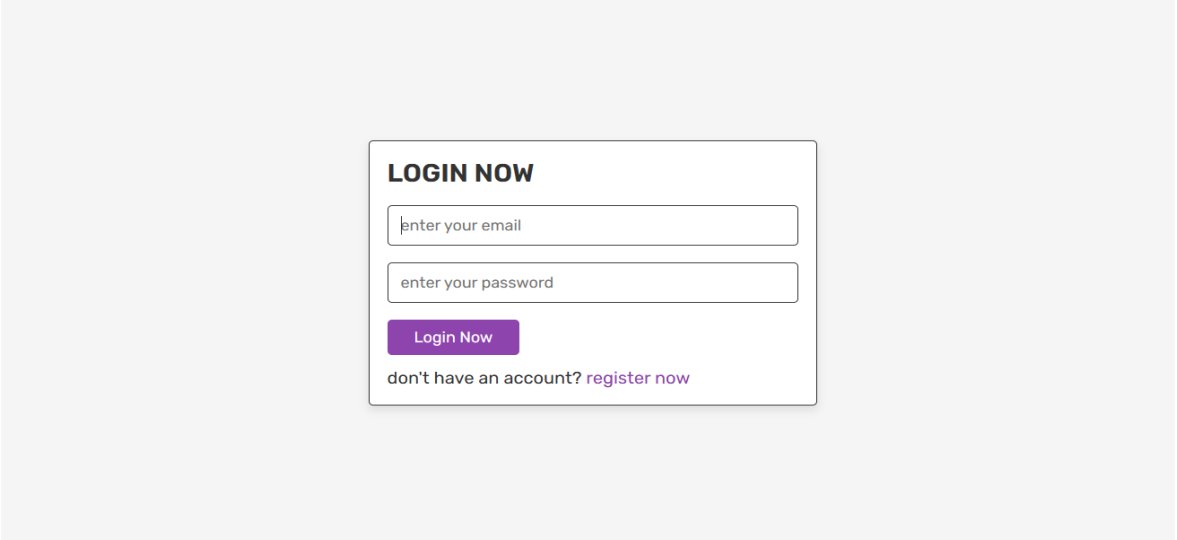

Figure A 2.1 Login Page

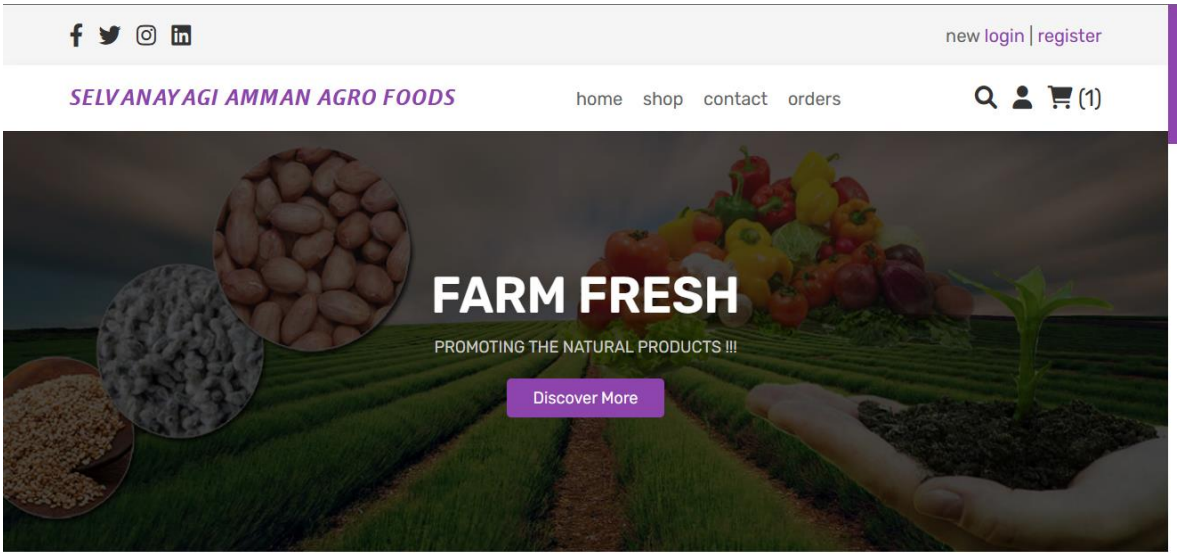

Figure A 2.2 Home Page

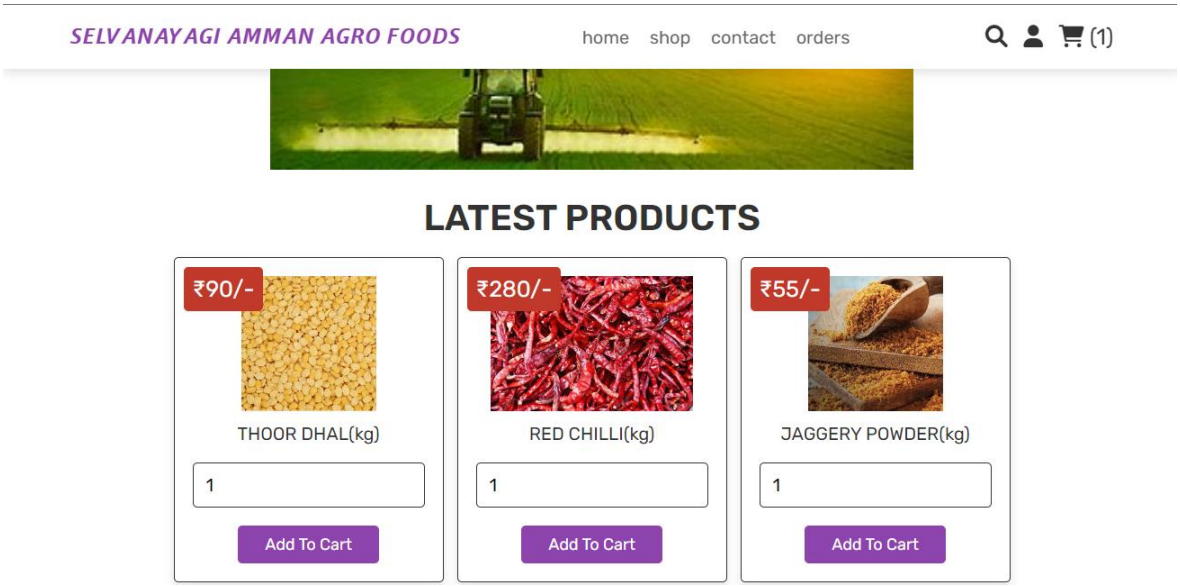

Figure A 2.3 Products Page

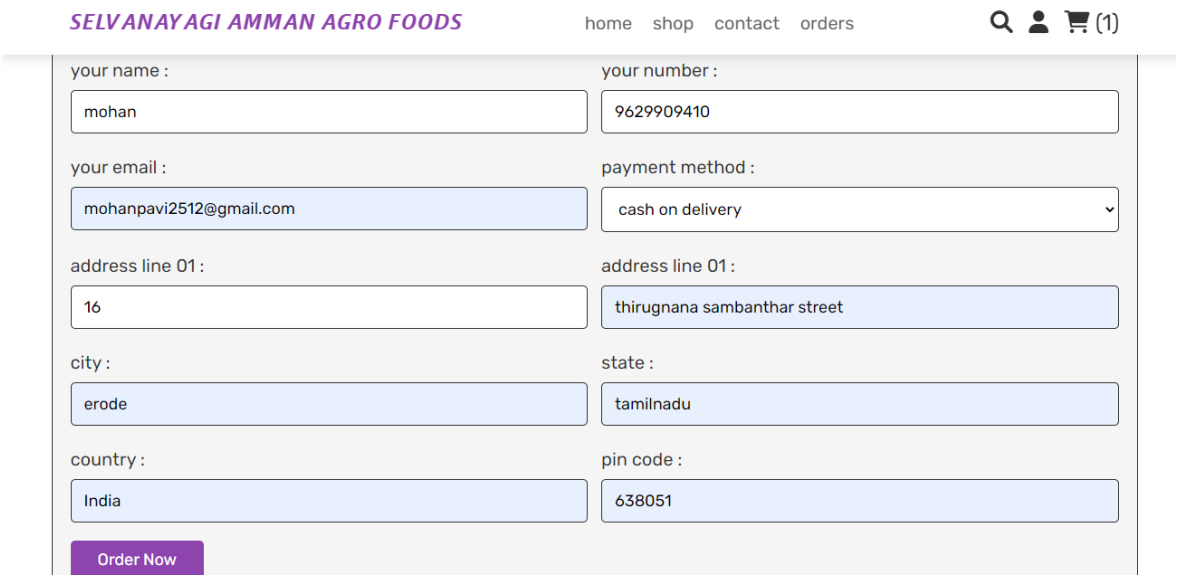

Figure A 2.4 Order Page

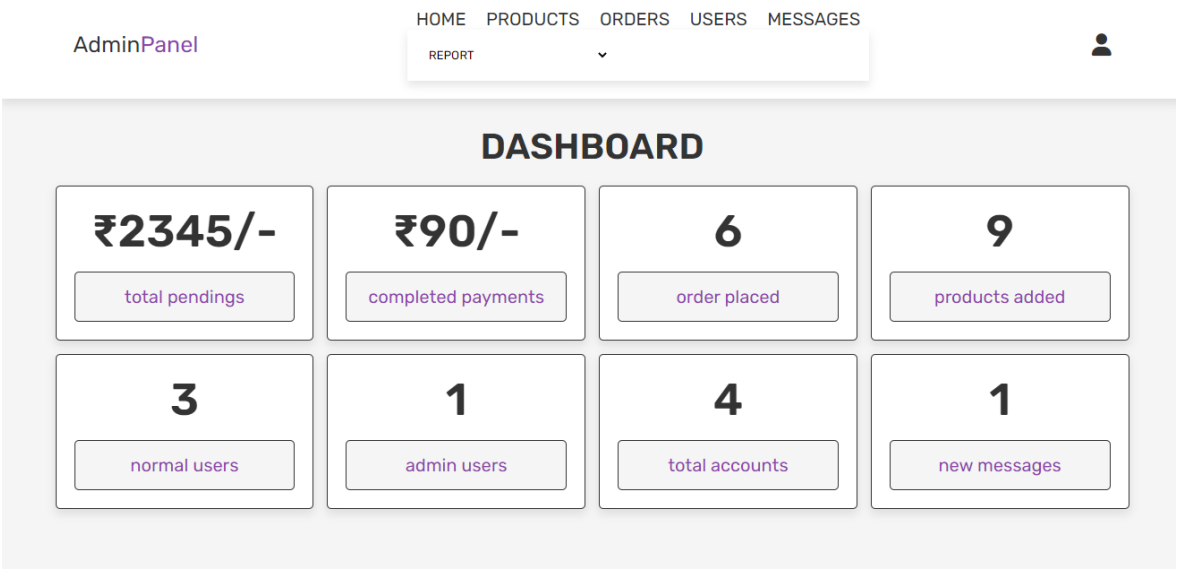

Figure A 2.5 Admin Dashboard

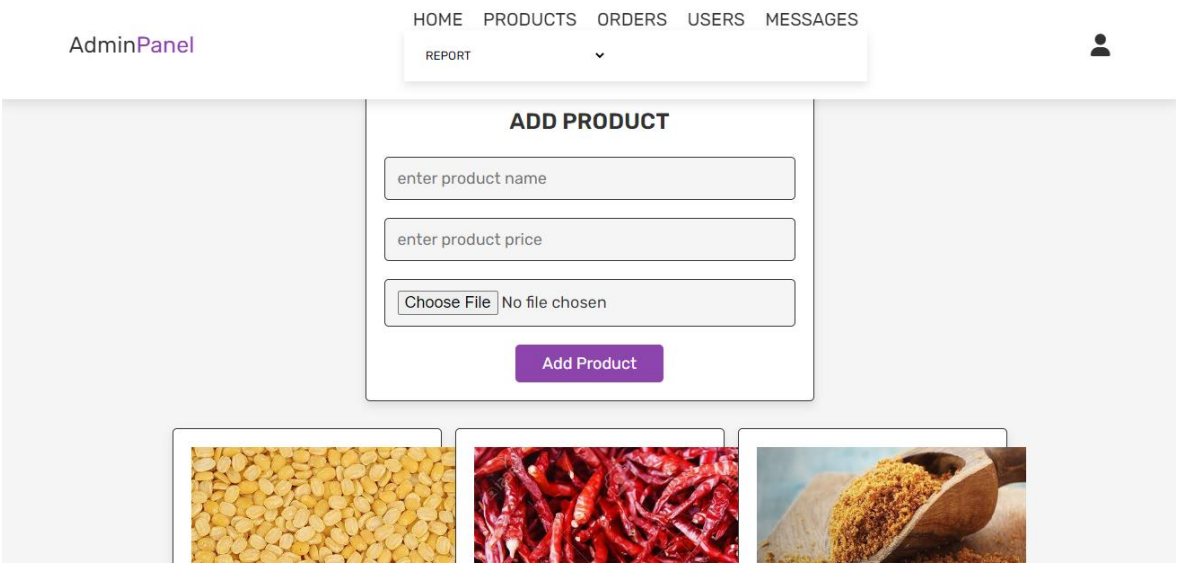

Figure A 2.6 Add Agro products

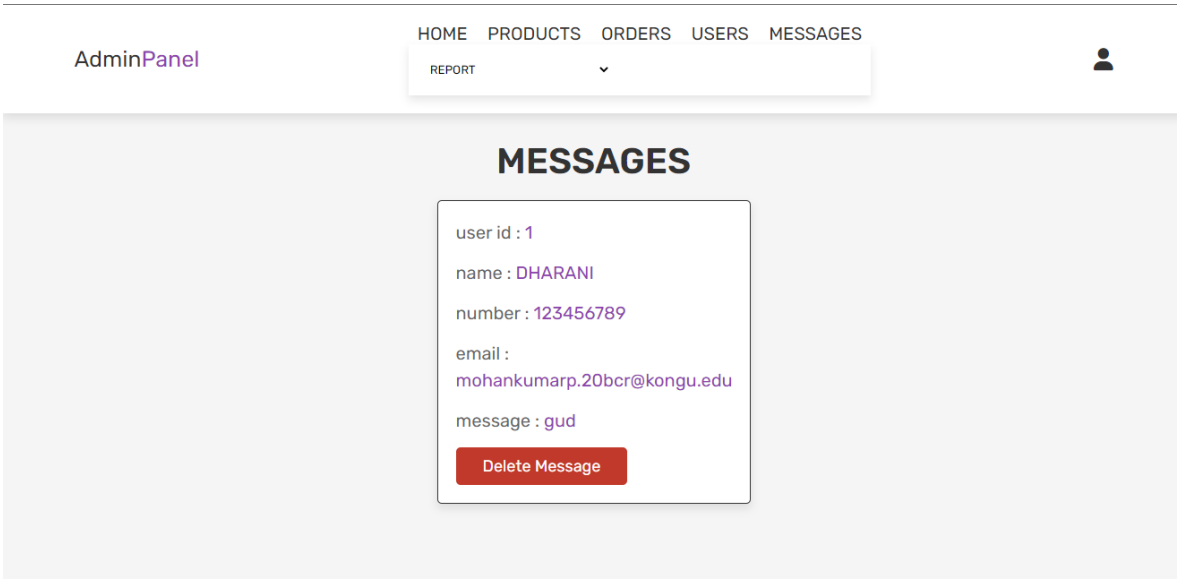

Figure A 2.7 Message Page

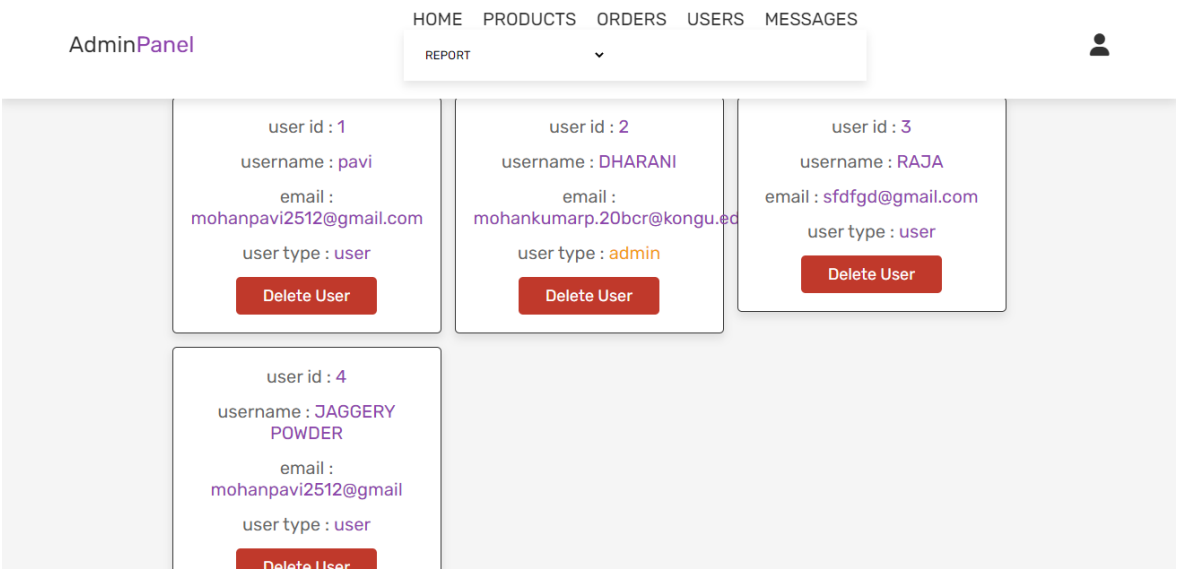

Figure A 2.8 Users Page

Order Date	Client Name	Contact	Grand Total
2022-11-01	pavi	57672686792	90
2022-11-15	pavi	57672686792	1135
2022-11-28	DHARANI kumar	57672686792	280
2022-11-28	RAJA	456789	370
2022-12-09	pavi	7894561230	800
Total Amount			2675

Figure A 2.9 Report Page

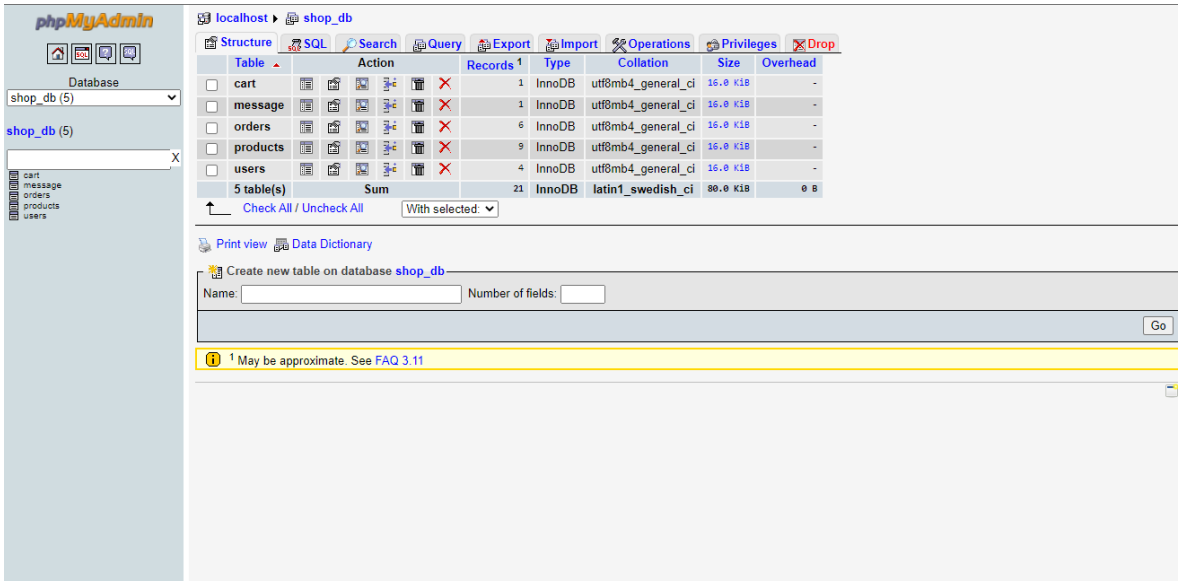

Figure A 2.10 Database Table

REFERENCES

[1] Kevin Yank (2015),'PHP &MYSQL Novice to Ninja' Fifth edition, Shroff Publishers &Distributers Private Limited.

[2] Luke Welling and Laura Thomson, "MYSQL Web Development-developer Library",2nd Edition.

[3] Luck Welling & Laura Thompson (2016),'PHP & MYSQL web Development' FifthEdition, Pearson Education.

[4] Mario Lurig (2017), 'PHP: Beginner's to Intermediate PHP5' First Edition, McGrawHill Education

[5] Robin Nixon, "Learning PHP, MYSQL & CSS & Html5"

[6] www.w3school.com,'Learn PHP and HTML'