INTRODUCTION

Geographical Indication (GI) products are a key component of many countries' agricultural, handicraft, and industrial heritage. These products derive their unique qualities and characteristics from the geographical region in which they are produced, and are protected under intellectual property rights law.

The online marketplace for GI products has been growing rapidly in recent years, providing consumers with easy access to high-quality, authentic products and offering producers the opportunity to expand their customer base and increase sales.

Developing an e-commerce website for Geographical Indication products requires a combination of technical expertise and knowledge of the unique characteristics of these products. The website must be user-friendly, secure, and capable of handling a large volume of transactions. In addition, it must provide detailed information about the products, their origins, and their unique qualities in order to educate and inform consumers.

This project will utilize various tools and techniques to create a robust and effective online shopping management system for Geographical Indication products. By developing an efficient and user-friendly platform, we aim to promote the consumption of GI products while also providing a valuable resource for producers and consumers alike.

1.1 About the Organization

Effective organization is key to the success of any e-commerce shopping platform. The important aspects of organization for an e-commerce shopping platform are:

1.Product organization: Organizing product is critical to providing a good user experience. Product should be categorized into logical groups and displayed in a way that makes it easy for users to find what they are looking for. This can include categories, sub categories, filters and search functionality.

2.Inventory management: It's important to have an effective inventory management system in place to ensure that products are available when customers want to purchase them. This can include real time inventory tracking, automated restocking, and notifications when inventory levels are low.

3.Order management: Efficient order management is essential to ensure that orders are processed and fulfilled quickly and accurately. This can include order tracking, shipping notifications and automated invoicing.

4.Customer relationship management: Effective customer relationship management is essential to building loyalty and repeat business. This can include tools for managing customer data, tracking customer interactions, and offering personalized promotions and discounts.

5.Payment and security: Providing secure payment options is essential to building trust with customers. This can include offering multiple payment methods, such as credit cards, pay pal, and apple pay, and using secure encryption to protect customer data.

6.Analytics and reporting: Tracking and analyzing user behavior and sales data can provide valuable insights into how to improve the e-commerce shopping platform. This can include tools for tracking website traffic, user engagement and conversion rates.

Effective organizations are critical to the success of an e commerce shopping platform. By focusing on product organization, inventory management, order management, customer relationship management, payment and security and analytics reporting, you can help ensure that your e commerce shopping platform is organized and optimized for success.

1.2 Objective

Developing a GUI based automated system, which will cover all the information Related to the all products which is used in our daily life. For example- Mobiles Phones, Laptops, Clothes, Books, Electronic Items and many more. So by this GUI based automated system a user wants to purchase something then it only a mouse clicks away to purchase these products.

The e-commerce is mainly useful for haven't time to go shopping for comfortably to the customers. Those are just entered into this website and bought they want at any time they can visit the web-site. Customer will choose different items like mobile, laptops, etc. This website is based on this formal. After chosen items they pay bill thorough pay pal process. Customers will get their items just sitting at home.

SYSTEM STUDY

System Study is an initial step for developing any new application software. A report will be prepared on the basis of a study conducted on the current process. Additional requirements and modifications are recommended for the purposed system. This phase of software development lays foundation for a detailed system analysis & design phase of proposed system. In practice the system study is done in two phases. In first phase, the preliminary survey of the system is done which helps in identifying the scope of the system. And in the second phase of the system study, the identification of the user's requirements & the limitations & problems of the present systems are studied

2.1 TOOLS AND TECHNIQUES USED TO COMPLETE THE STUDY

The development of the Geographic Indication (GI) products e-commerce website with PHP will require a range of tools and techniques, including:

• **IDE:** An integrated development environment, such as Eclipse or NetBeans, will be used to write, debug and test PHP code.

• **PHP Frameworks:** Web development frameworks like Laravel or CodeIgniter will be used to simplify and accelerate the development process.

• **Database Management Systems:** A database management system such as MySQL will be used to store and manage data, including product listings, user accounts, and transaction records.

• **Front-end Development Tools:** HTML, CSS, and JavaScript will be used to design and develop the website's user interface.

• **Geographic Information Systems:** GIS software will be used to manage and analyze geographical data, including information about the origin and production process of GI products.

• **Quality Assurance Tools:** Testing frameworks, bug tracking systems, and code review tools will be used to ensure that the website is stable, secure, and free from errors.

• **Project Management Tools:** Project management tools like Trello or Asana will be used to manage and coordinate the development process, assign tasks, and track progress.

2.2 PROBLEMS IN THE EXISTING SYSTEM

The existing system for buying and selling GI products is typically offline or conducted through physical marketplaces, which can be time-consuming and limited in reach. The lack of an online platform makes it difficult for sellers to showcase their products to a wider audience, and for buyers to discover and purchase products from different regions.

2.3 SOLUTION TO THE PROBLEM

The Geographic Indication (GI) products e-commerce website with PHP will provide a convenient and secure platform for buying and selling GI products. The website will feature an intuitive and user-friendly interface with search and filter options that allow buyers to easily find and purchase products from different regions. Sellers will have the ability to showcase their products to a wider audience and manage their inventory through a centralized database. The website will also feature a secure payment system that enables buyers to pay for their purchases using various payment methods. Overall, the website will streamline the process of buying and selling GI products, making it more efficient and accessible for both buyers and sellers.

SYSTEM ANALYSIS

The objective of the system analysis activity is to develop structured system specification for the proposed system. The structured system specification should describe what the proposed system would do; independent of the technology, which will be used to implement these requirements. The structured system specification will be called the essential model also known as logical model.

The essential model may itself consist of multiple models, modelling different aspect of the system. The data flow diagrams may model the data and their relationships and the state transition diagram may model time dependent behavior of the system. The essential model thus consists of the following:

- Context diagram
- Leveled data flow diagram
- Process specification for elementary bubbles
- Data dictionary for the flow and stores on the DFDs.

3.1 Problem Analysis

The problem analysis phase involves identifying the problems that the current online shopping system faces and the solutions to overcome them. The problems identified include the lack of a user-friendly interface, slow loading times, and the inability to manage orders efficiently. Ecommerce system is a computerized, online solution to the various problems faced by the product buyer and seller wishing to outsource their software development work to a provider at an economical cost, thus achieving high performance, accuracy, reliability, and high speed of data retrieval.

In this system, there is a registration process each for the Product buyer and seller. The Administrator of the site verifies the Provider after his registration and if satisfied, assigns him a user name and password.

Our site can be used by anyone who is searching for Products whether he/she is first time visiting our site. Our site also provides some discounted products as same u get on any shop.

The software covers the following point while keeping in mind user's requirement:

- Fast online access of information about various products.
- Search products by keywords like functional area, experience and also by initials of the product's name.

Administrator will maintain the database and perform all process.

These are 2 categories of users-

1. General User
2. Registered User

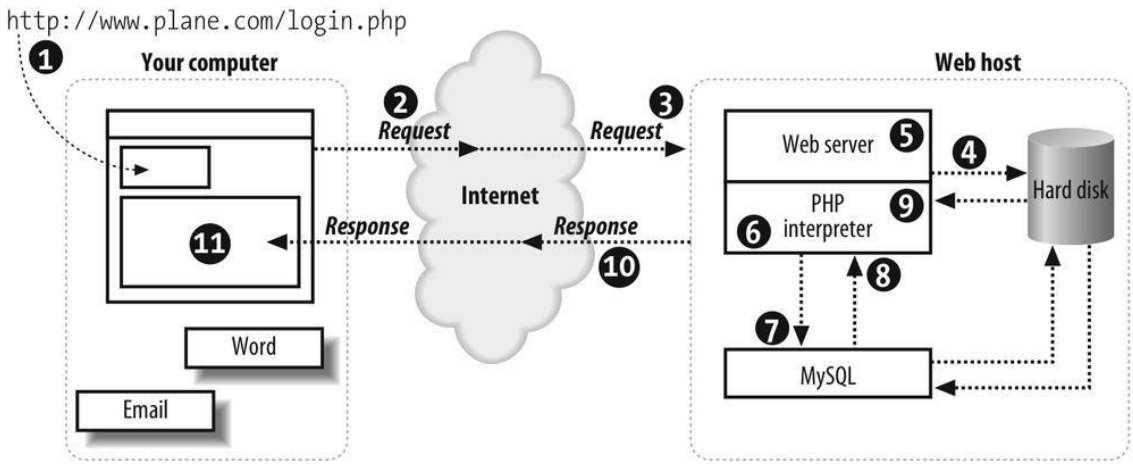
3.2 Feasibility Study

An important outcome of the preliminary investigation is the determination that the system requested is feasible. Feasibility study is carried out to select the best system that meets the performance requirements. It begins with a request from the user for a new system. It involves the following:

- ❖ Identify the responsible user for a new system
- ❖ Clarify the user request
- ❖ Identify deficiencies in the current system
- ❖ Establish goals and objectives for the new system
- ❖ Determine the feasibility for the new system
- ❖ Prepare a project charter that will be used to guide the remainder of the project

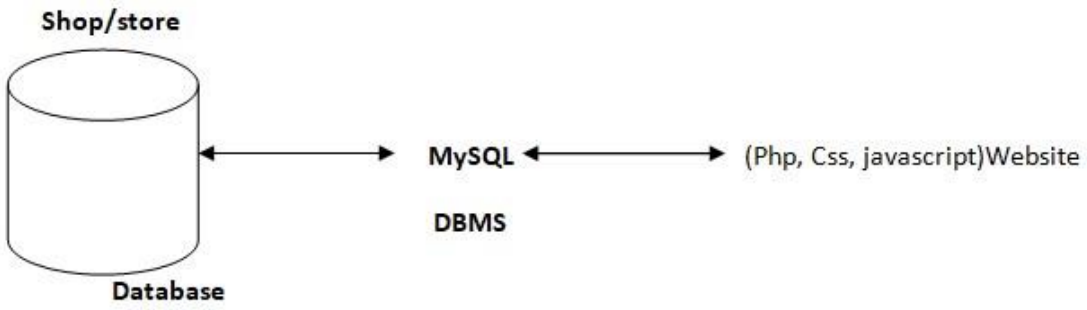
3.3 Tools and Environment used

1. PHP – PHP is a server-side scripting language that is used to develop Static websites or Dynamic websites or Web applications. It is designed for web development to implement dynamic web pages and can be embedded into HTML for it to be displayed

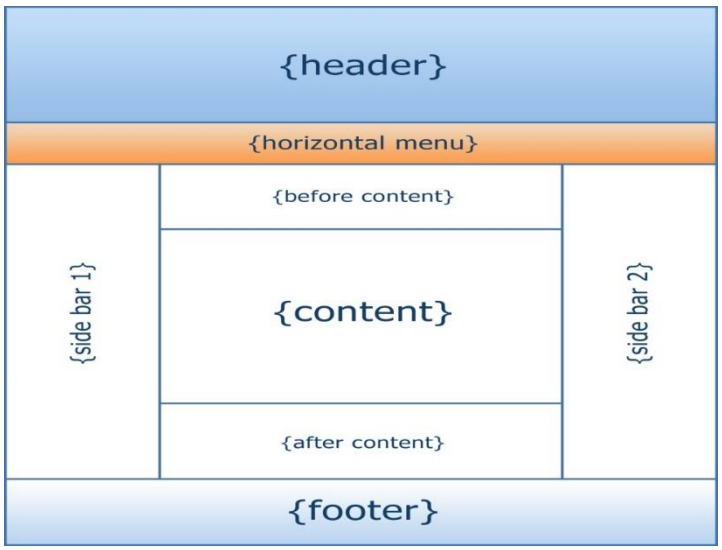


2. **Xampp –** XAMPP is a free open source cross-platform web server solution stack package developed by Apache friends, consisting mainly of the Apache HTTP server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache(A), Maria DB(M), PHP(P), and Perl(P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purpose. Everything needed to set up a web server- server application (Apache), database (Maria DB), and scripting language (PHP)- is included in an extractable file. XAMPP works equally well on Linux, Mac and windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

3. **MYSQL –** MySQL is a free source database system, and it enables the cost-effective delivery of reliable and a high-performance and scalable Web-based and embedded database applications. It is a relational database system (RDBMS). It is a high performing program and scalable to meet the demands of users and data. MySQL is written in C and C++, so it is compatible with most of the operating systems available around the world.



4. **HTML –** Hypertext Markup Language. This language is used in creating web pages. This language also supports other languages such CSS, PHP, JAVASCRIPT, etc. in creating interactive and responsive pages on the pages. HTML5 is just an updated version of the HTML. It supports new features, new attributes, new HTML elements, full CSS3 support, video and audio, 2D/3D graphics that help users and also help web developers to create new features easily on the website. The structure of HTML5 is shown in figure



5. **SUBLIME TEXT 3:** Sublime text 3 (ST3) is a lightweight, cross-platforms code editor known for its speed, ease of use, and strong community support. It's an amazing editor that works out of the box, but its real power lies in the ability to extend its functionality by controlling packages and creating custom configurations.

6. **CSS:** CSS is simply referred to as Cascading Style Sheets.CSS is used to define styles for web pages, including the design, layout, and variations in the display for different devices and screen sizes. /3/ The general structure of CSS Basic syntax: selector {property: value}
   The value you want the property to take

   Example:
   a. *p {text-align: center; color: black; font-family: Arial}*

   CSS can be used in a separate style sheet or used in the webpage

```
<! DOCTYPE html>

<html lang = "en-US">

<head>

<meta charset = "UTF-8">

<link rel = "stylesheet"  type = "text/css"  href =

"myStyle.css" />

</head>
```

7. **JAVASCRIPT:** JavaScript is a high-level language which could be used independently or inculcated into the webpage. It can be used to, handle requests and responses and also add dynamic behavior and also store information on a website.

```
<script ="javascript" type="text/javascript">

JavaScript code

</script>
```

8. **BROWSER:** A browser is a software program that is used to explore, retrieve, and display the information available on the World Wide Web. This information may be in the form of pictures, web pages, videos, and other files that all are connected via hyperlinks and categorized with the help of URLs (Uniform Resource Identifiers).

9. **BOOTSTRAP:** Bootstrap is a popular front-end web development framework. It was created by Twitter in 2010 to standardize web development across teams. Bootstrap is based on HTML, CSS, and JavaScript and includes pre-designed templates and components. It uses a 12-column grid system to create responsive layouts for different screen sizes. Bootstrap supports a wide range of browsers and devices. Bootstrap is free and open source with a large community of developers.

10. **GITHUB:** GitHub is a popular web-based platform for hosting and managing software development projects. GitHub was founded in 2008 and acquired by Microsoft in 2018. It uses Git, a distributed version control system, to manage and track changes to code. GitHub provides a web-based interface for creating and managing repositories, which are used to store code, documentation, and other project assets. GitHub allows developers to collaborate on projects and contribute code through pull requests.

11. **STACKOVERFLOW:** Stack Overflow is a popular online community where developers and programmers can ask and answer technical questions related to software development, programming, and computer science. It was founded in 2008 and has become one of the go-to resources for developers to seek help and share knowledge.

    Users can post their questions on Stack Overflow and receive answers and suggestions from other users around the world. The community is known for its active moderation and high-quality content, with answers being upvoted or down voted based on their relevance and usefulness.

3.4 Hardware Configuration

Processor	: Inter(R) Core (TM) i5-52000
CPU Speed	: 2.20GHz
RAM	: 8.00 GB
Hard Disk	: 500 GB
Monitor	: 15.6”

INPUT DEVICE

Key board	: 101 Keys
Mouse	: optical mouse

3.4 SOFTWARE CONFIGURATION

Operating system	: Window 10
Front End	: PHP
Back End	: SQL Server

SYSTEM DESIGN

System design involves transformation of the user implementation model into software design. The design specification of the proposed system consists of the following:

- Database scheme
- Structure charts
- Pseudo codes for the modules in structure charts

DATABASE SCHEME

A database scheme is a structured that defines the organization of data in a database system. It includes tables, columns, data types, and relationships between tables. In system design, a well-designed database scheme is critical for the success of the system, as it ensures efficient data storage, retrieval, and processing.

A database scheme should be designed to support the requirements of the system, including its data storage and retrieval needs, performance requirements, and scalability. It should also consider the security and privacy requirements of the system, such as data encryption, access control, and auditing.

A good database scheme design should be normalized to minimize data redundancy and ensure data consistency and integrity. Normalization involves breaking down a table into smaller tables to reduce data duplication and improve data integrity. Overall, a database scheme is an essential components of system design, and it should be carefully planned and implemented to ensure that the system can effectively manage its data.

STRUCTURE CHARTS

Structure charts are a popular tool used in system design to represent the hierarchical structure of a computer program or system. They are used to depict the organization and flow of processes, functions, and data within a system.

A structure chart typically consists of a series of rectangles, representing modules or processed, connected by arrows, indicating the flow of data between them. The modules are arranged in a hierarchy, with higher-level modules controlling the lower-levels ones. The structure chart provides a visual representation of the program's structure, allowing the designer to easily identify modules and functions that need to be created and how they relate to each other.

There are several key elements of a structure chart:

1. **Modules or Processes:** These are the rectangular boxes that represent the individual processes or functions within the system.
2. **Connecting arrows:** These arrows connect the modules and represent the flow of data or control between them.
3. **Annotation:** These are labels that provide information about the purpose and function of each module.
4. **Hierarchy:** The modules are arranged in a hierarchical structure, with higher-level modules controlling the lower-level ones.
5. **Levels:** Each module is assigned a level indicating its position in the hierarchy, with level 0 representing the main module or program.

Structure chart are useful for breaking down complex systems into smaller, more manageable components. They can be used to plan and organize the development of large software system, and to communicate the structure and organization of a system to other members of a development team.

PSEUDO CODES FOR THE MODULES IN STRUCTURE

Pseudo code is a high-level language that allows developers to outline the basic logic of a program or module without getting bogged down in the specifies of any particular programming language. In system design, structure charts are often used to visualize the different modules that make up the system, and pseudo code can be used to describe the behavior of each module.

1. **Pseudocode should be clear and concise:** The purpose of pseudocode is to provide an overview of the logic of a module. As such, it should be easy to read and understand by both technical and non-technical stakeholders.

2. **Pseudocode should be language- agnostic:** Since pseudocode is not tied to any particular programming language, it should avoid using specific language constructs or syntax, Instead, it should use plain language that is easy to understand.

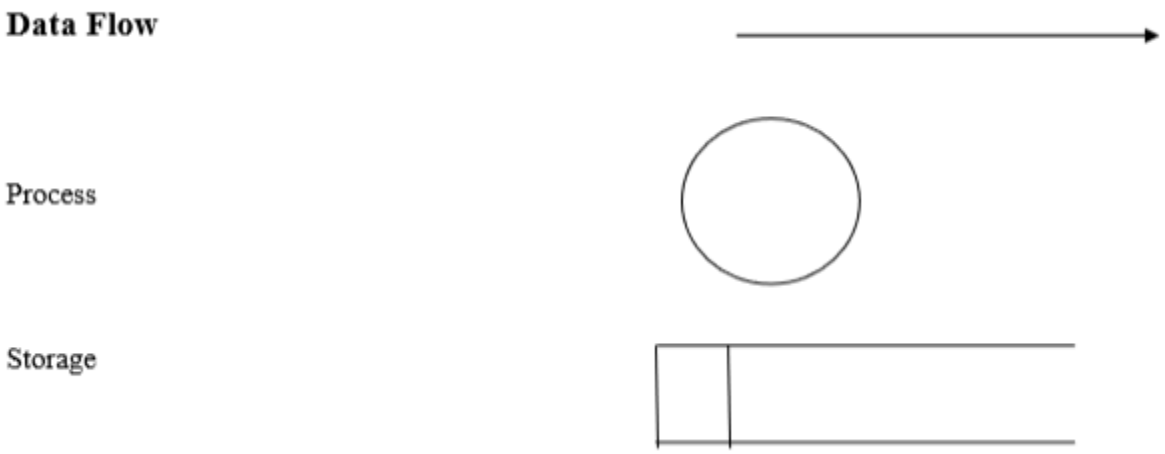
3. **Pseudocode should be modular:** Each module in a structure chart should have its own pseudocode, which should focus on describing the module's specific behavior and interactions with other modules in the system.

4. **Pseudocode should be based on the design specifications:** The Pseudocode for each module should be based on the design specifications and requirements for that module. It should describe how the module should behave in different scenarios and how it should interact with other modules in the system.

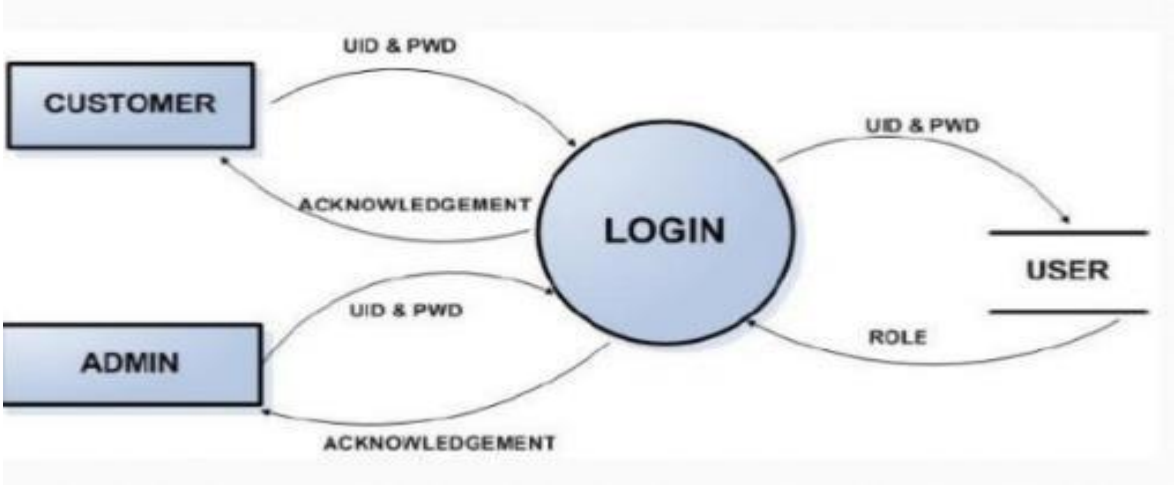
5. **Pseudocode should be reviewed and tested:** Once the pseudocode for each module has been written, it should be reviewed and tested to ensure that it accurately reflects the intended behavior of the module. This can help identify potential issues or errors before the module is actually implemented.

4.1 DATA FLOW DIAGRAM

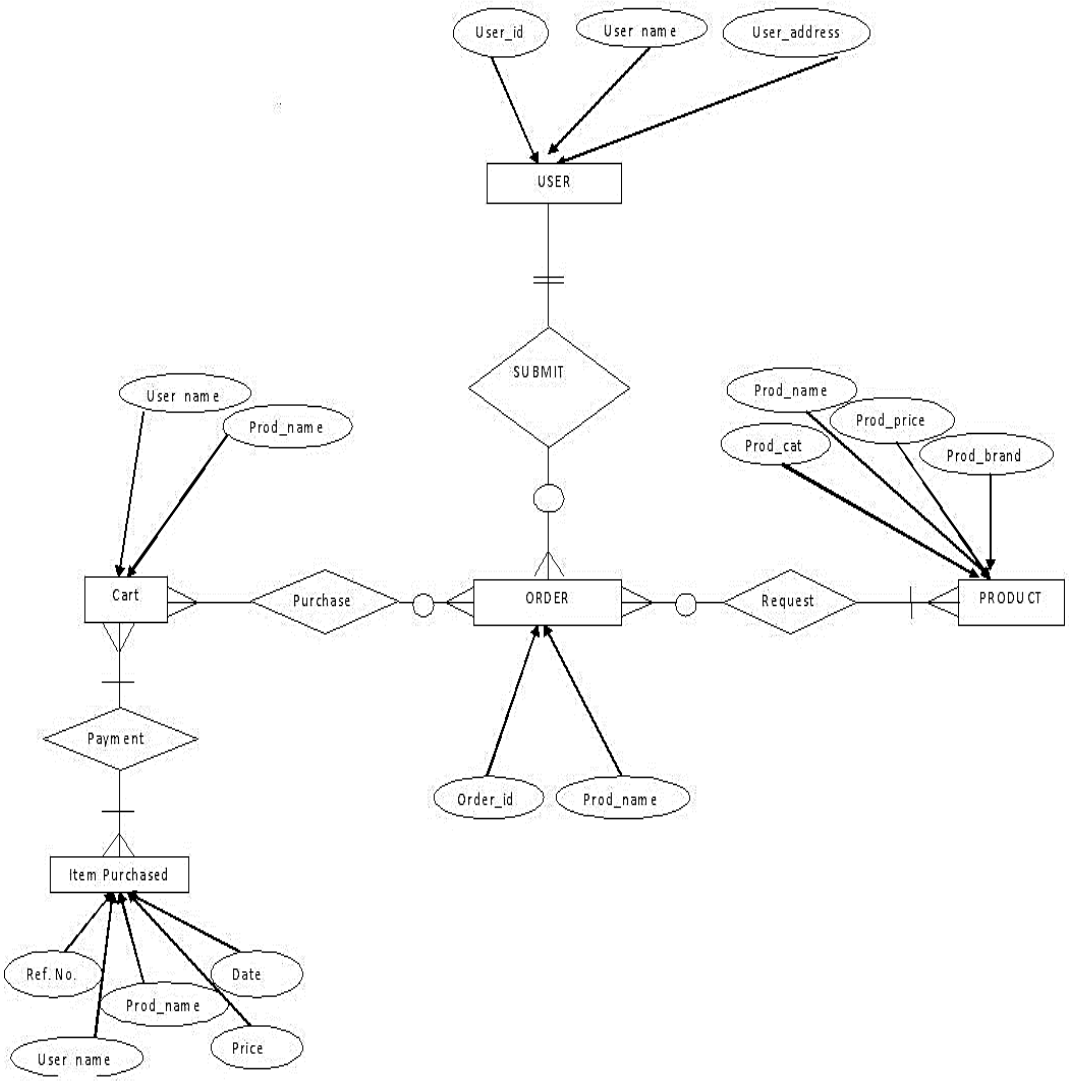
Data flow diagram are directed graph in which the modes specify processing activities and the arcs specify data item transmitted between processing nodes. The data flow diagrams that are used for the design of proposed system are given below:



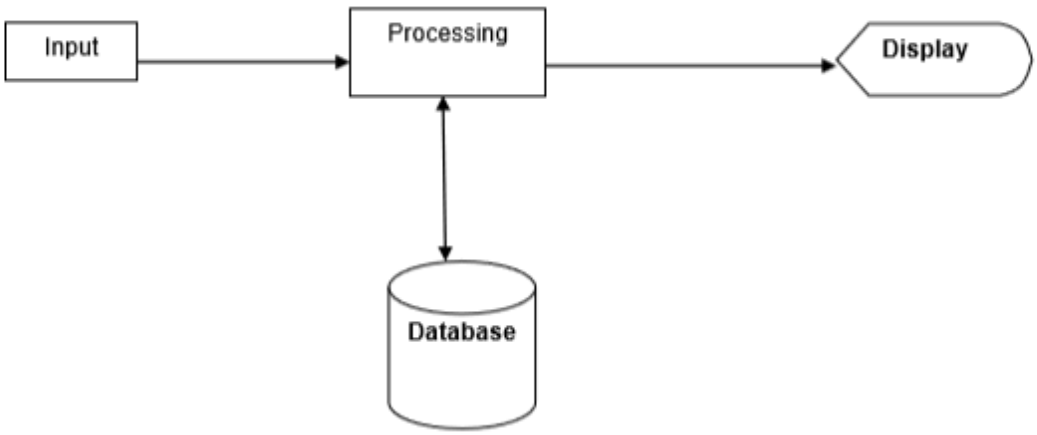
The data flow diagrams that are used for the design of proposed system are given below:



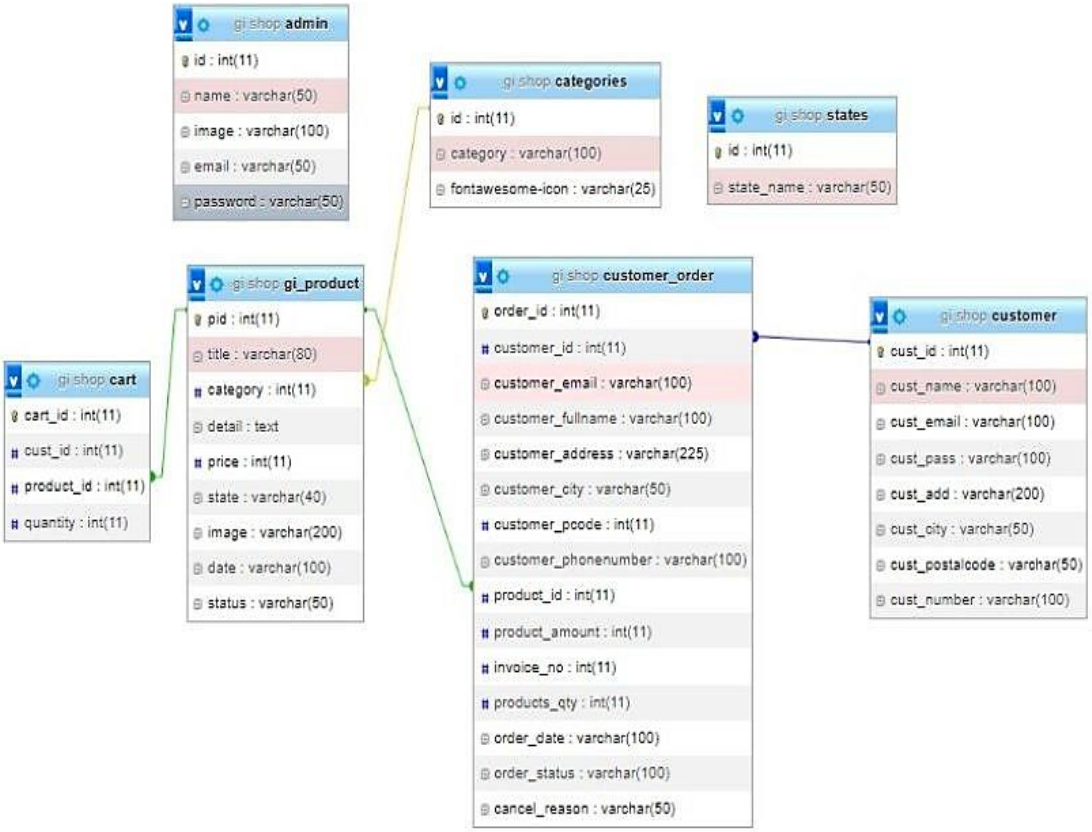
SYSTEM DATAFLOW



System Diagram



4.2 ENTITY RELATIONSHIP DIAGRAM



for each table, we need to mark at the least one area as a primary key. The primary key is usually specified in a particular table. The table "admins" is created to store all the Admin' information. It consists of eight distinctive fields referred to as columns to shop admin id, first name, last name, email, and password. The email and the password are essential because they are needed for one to login into the shop. "*admin_id*" is the primary KEY. Many columns of one table may be described as primary KEY. Each column has a name, a datatype and different nonobligatory attributes.

4.3 Input Design

Input design is the process of converting user-oriented input to a computer-based format. The input data were collected and once they were identified, appropriate input media were selected for processing. The goal for designing input was to make data entry as easy and free from errors as possible. Additionally, the input design phase makes sure that data item and transaction have validation to detect errors and comment to warn the users for invalid input. Menus are providing for easier access to data and to simplify the job of screens for proposed system has been designed in such a way so as to incorporate all the validation aimed at providing and user friendliness.

4.4 Output Design

Computer output is the most important and direct source of information to user. Efficient & intelligent output design improves the system relationship with the user and help in decision-making. The output provides a means of storing a copy of result for later reference and consultation. Output generally refers to the result and it can be in the form of operational document or report.

4.5 Database Design

Designing a database is carried out with proper planning which require team effort and corporation. Improper database design, if implemented may result in too few or too many fields. Data base design describes how the data should be stored and how the users will ask for data from the database. Database design runs in parallel with the application design. Various step involved in the database design are listed below:

- **Data definition**

  In the first phase, we must make lists of all attributes involve in our application. It should be noted that, all the fields of our application might be required, should be included.

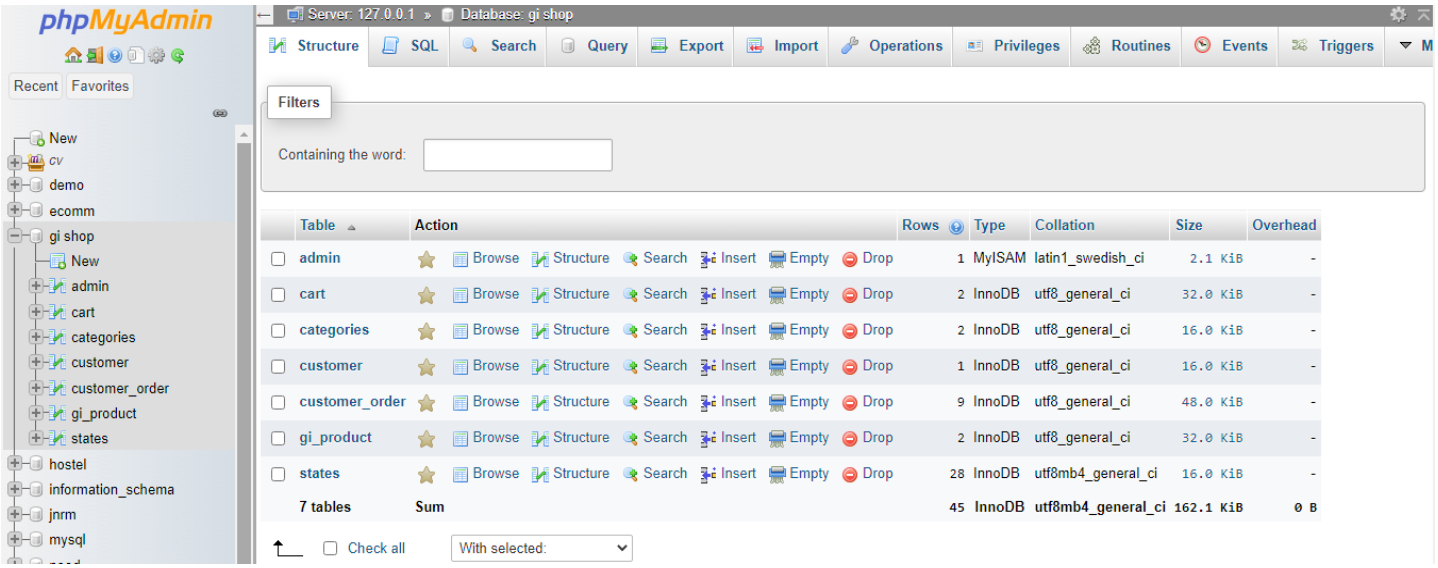
- **Data Refinement**

  The second step in the database design is to refine our initial list of fields so that the fields from the accurate description of the type of data will be needed in the database.

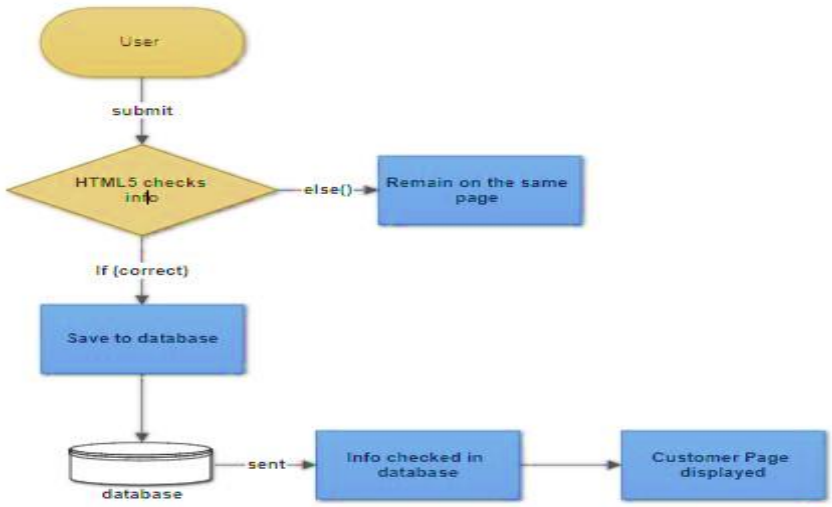
- **Identifying the keys**

  The next step in the database design is to identify the key that is used as a primary key. A primary key is a key which uniquely identifies a row in a table. The foreign key is identified. A foreign key is a key which acts as a primary key for another table.

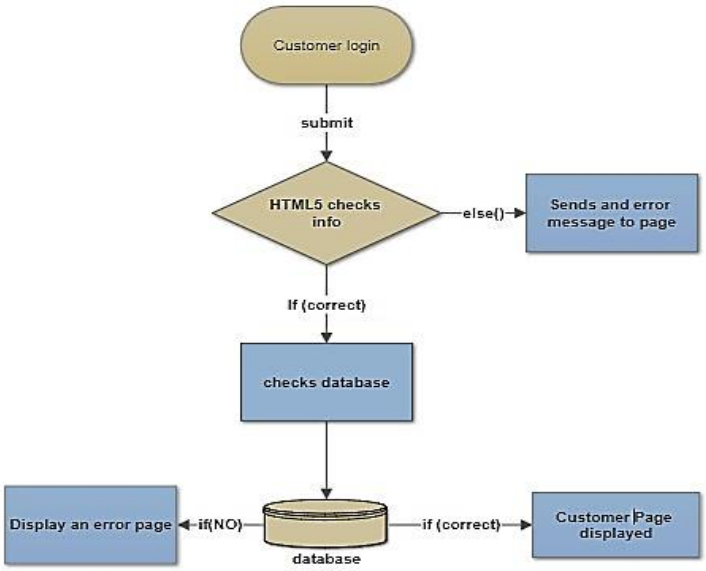
  - **Database Table**



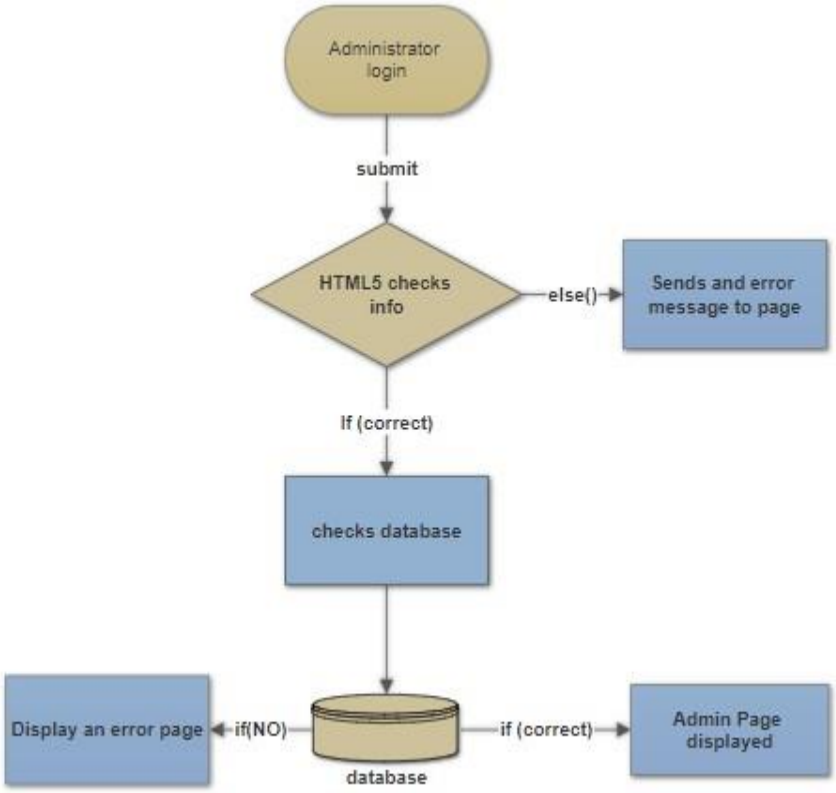
  - **User registration**



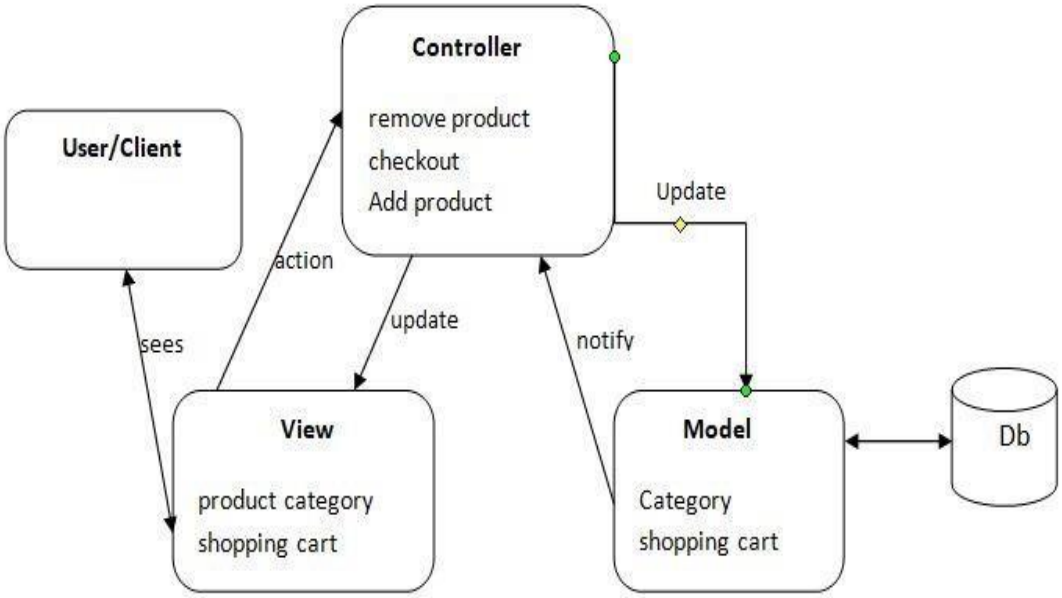
- **Customer login**

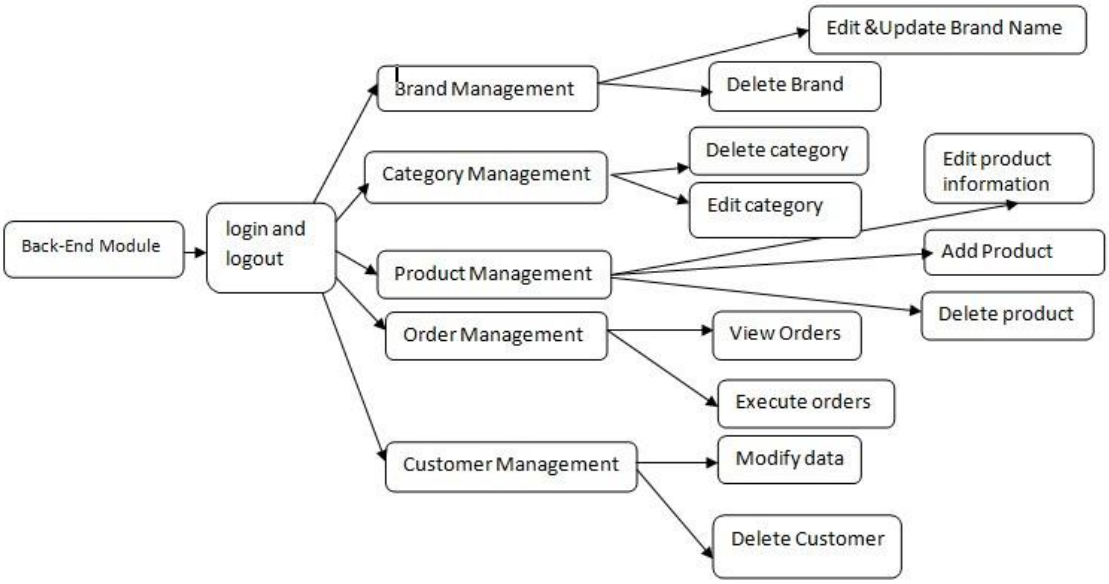


- Administration Login



- **Backend dataflow**





SAMPLE CODING

Connecting With database

```
<?php
 $host      = "localhost";
 $username  = "root";
 $passwrod  = "";
 $dbName    = "gi shop";
  $con = mysqli_connect($host, $username,$password,$dbName);
   ?>
```

SQL DATABASE

Database: `gi shop`

Table structure for table `admin`

```
CREATE TABLE `admin` (
  `id` int (11) NOT NULL,
  `name` varchar (50) NOT NULL,
  `image` varchar (100) NOT NULL,
  `email` varchar (50) NOT NULL,
  `password` varchar (50) NOT NULL
)
```

Table structure for table `cart`

```
CREATE TABLE `cart` (
  `cart_id` int (11) NOT NULL,
  `cust_id` int (11) NOT NULL,
  `product_id` int (11) NOT NULL,
  `quantity` int (11) DEFAULT NULL
)
CREATE TABLE `categories` (
  `id` int (11) NOT NULL,
  `category` varchar (100) CHARACTER SET latin1 NOT NULL,
  `fontawesome-icon` varchar (25) CHARACTER SET latin1 NOT NULL
)

CREATE TABLE `customer` (
  `cust_id` int (11) NOT NULL,
  `cust_name` varchar (100) NOT NULL,
  `cust_email` varchar (100) NOT NULL,
  `cust_pass` varchar (100) NOT NULL,
  `cust_add` varchar (200) NOT NULL,
  `cust_city` varchar (50) NOT NULL,
  `cust_postalcode` varchar (50) NOT NULL,
  `cust_number` varchar (100) NOT NULL
)
```

```
CREATE TABLE `customer_order` (
  `order_id` int (11) NOT NULL,
  `customer_id` int (11) NOT NULL,
  `customer_email` varchar (100) NOT NULL,
  `customer_fullname` varchar (100) NOT NULL,
  `customer_address` varchar (225) NOT NULL,
  `customer_city` varchar (50) NOT NULL,
  `customer_pcode` int (11) NOT NULL,
  `customer_phonenumber` varchar (100) NOT NULL,
  `product_id` int (11) NOT NULL,
  `product_amount` int (11) NOT NULL,
  `invoice_no` int (11) NOT NULL,
  `products_qty` int (11) NOT NULL,
  `order_date` varchar (100) NOT NULL,
  `order_status` varchar (100) NOT NULL,
  `cancel_reason` varchar (50) NOT NULL
)
Table structure for table `gi_product'
CREATE TABLE `gi_product` (
  `pid` int (11) NOT NULL,
  `title` varchar (80) NOT NULL,
  `category` int (11) NOT NULL,
  `detail` text NOT NULL,
  `price` int (11) NOT NULL,
  `state` varchar (40) NOT NULL,
  `image` varchar (200) NOT NULL,
  `date` varchar (100) NOT NULL,
  `status` varchar (50) NOT NULL
)
-- Table structure for table `states`
CREATE TABLE `states` (
  `id` int (11) NOT NULL,
  `state_name` varchar (50) NOT NULL
)
```

DASHBOARD

```
<?php include('include/header.php');
if(isset($_SESSION['email'])) {
  $custid = $_SESSION['id'];
  if(isset($_GET['cart_id'])){
    $p_id = $_GET['cart_id'];
    $sel_cart = "SELECT * FROM cart WHERE cust_id = $custid and product_id = $p_id
";
    $run_cart    = mysqli_query($con,$sel_cart);
    if(mysqli_num_rows($run_cart) == 0){
     $cart_query = "INSERT INTO `cart`(`cust_id`, `product_id`,quantity) VALUES
($custid,$p_id,1)";
      if(mysqli_query($con,$cart_query)){
       header('location:index.php');
      }
    }
    if(mysqli_num_rows($run_cart) > 0){
     while($row = mysqli_fetch_array($run_cart)){
       $exist_pro_id = $row['product_id'];
         if($p_id == $exist_pro_id){
          $error="<script> alert('⚠ This product is already in your cart  ');</script>";
         }
       }
      }
    }
  }
  else if(!isset($_SESSION['email'])){
   echo "<script> function a(){alert('⚠ Login is required to add this product into
cart');}</script>";
  }
?>
      <!--Carousel Wrapper-->
```

```
<div class="carousel slide mt-5" id="slider" data-ride="carousel" >
    <!---Indicators-->
    <ol class="carousel-indicators">
      <li data-target="#slider" data-slide-to="0" class="active"></li>
      <li data-target="#slider" data-slide-to="1" ></li>
      <li data-target="#slider" data-slide-to="2" ></li>
    </ol>
    <div class="carousel-inner" style="margin-top:10%">
      <div class="carousel-item active">
        <img src="img/spice.jpg" class="d-block w-100">
      </div>
      <div class="carousel-item">
        <img src="img/bamboo.jpg" class="d-block w-100">
      </div>
      <div class="carousel-item">
        <img src="img/bg.jpeg" class="d-block w-100">
      </div>
      <!---Controlers-->
      <a class="carousel-control-prev" data-slide="prev" href="#slider">
        <span class="carousel-control-prev-icon"></span>
      </a>
      <a class="carousel-control-next" href="#slider" data-slide="next" >
        <span class="carousel-control-next-icon"></span>
      </a>
    </div
  </div>
<section >
  <div class="container pt-5 pb-5">
  <div >
    <?php
    if(isset($msg)){
      echo $msg;
     }
    else if(isset($error)){
```

```
            echo $error;
           }
         ?>
        </div>
        <h1 class="text-center">Latest Products</h1>
        <div class="row mt-4">
          <?php
           $p_query = "SELECT * FROM gi_product ORDER BY pid DESC LIMIT 4";
           $p_run   = mysqli_query($con,$p_query);
           if(mysqli_num_rows($p_run) > 0 ){
              while($p_row = mysqli_fetch_array($p_run)){
               $pid      = $p_row['pid'];
               $ptitle  = $p_row['title'];
               $pcat    = $p_row['category'];
               $p_price = $p_row['price'];
               $state    = $p_row['state'];
               $img1    = $p_row['image'];
             ?>
             <div class="col-md-3 mt-3">
                <img src="img/<?php echo $img1; ?>" class="hover-effect" width="100%" height="190px">
                <div class="text-center mt-3">
                 <h5 title="<?php echo $ptitle;?>"><?php echo substr($ptitle,0,20); ?>...</h5>
                  <h6>Rs. <?php echo $p_price; ?></h6>
                </div>
                 <div class="row">
                    <div class="col-md-12 col-sm-12 col-12 text-center">
                    <a href="index.php?cart_id=<?php echo $pid;?>" type="submit" onclick="a()" class="btn btn-primary btn-sm hover-effect">
                       <i class="far fa-shopping-cart"></i>
                    </a>
                    <a href="product-detail.php?product_id=<?php echo $pid;?>" class="btn btn-default btn-sm hover-effect text-dark" >
```

```
                        <i class="far fa-info-circle"></i> View Details
                    </a>
                    </div>
                </div>
            </div>
            <?php
                }
            }
        else{
            echo "<h3 class='text-center'> No Product Available Yet </h3>";
        }
        ?>
    </div>
</div>
</section>
    <section class="back-gray pt-4 pb-4">
<div class="container">
    <h2 class="text-center">How It Does Work</h2>
    <div class="row">
        <!--choose product card-->
        <div class="col-md-4 p-5">
            <div class="card hover-effect" id="border-less">
                <div class="card-body mt-3 text-center">
                    <i class="fal fa-phone-laptop fa-3x"></i>
                    <div class="heading mt-2">
                        <h4>Product</h4>
                        <h6 class="text-secandary">Choose your own product</h6>
                    </div>
                    <p class="mt-2">Add product to cart & proceed to checkout and put your shipping address then checkout that. </p>
                </div>
            </div>
        </div>
        <!--cash on deliver-->
```

```
            <div class="col-md-4 p-5">
              <div class="card hover-effect" id="border-less">
                <div class="card-body mt-3 text-center">
                  <i class="fal fa-hand-holding-box fa-3x"></i>
                    <div class="heading mt-2">
                      <h4>Recieve</h4>
                      <h6 class="text-secandary">Recieve Your Product</h6>
                    </div>
                    <p class="mt-2">After chouckout from cart your product will be deliver at your door step within 7 working days</p>
                </div>
              </div>
            </div>
            <div class="col-md-4 p-5">
              <div class="card hover-effect" id="border-less">
                <div class="card-body mt-3 text-center">
                  <i class="fal fa-wallet fa-3x"></i>
                    <div class="heading mt-2">
                      <h4>Cash</h4>
                      <h6 class="text-secandary">Cash on delivery</h6>
                    </div>
                    <p class="mt-2">On delivery of your product hold or recieve your products then pay cash on that moment.</p>
                </div>
              </div>
            </div>
            <!---end cash on delivery-->
          </div>
        </div>
    </section>
<?php include('include/footer.php'); ?>
```

PLACE ORDER

```
<?php include('include/header.php'); ?>

      <div class="jumbotron">
         <h2 class="text-center mt-5">Checkout</h2>
      </div>

   <div class="container">
<?php
  if(isset($_SESSION['id'])){
     $customer_id    = $_SESSION['id'];
     $customer_email = $_SESSION['email'];
     $customer_name  = $_SESSION['name'];
     $customer_add   = $_SESSION['add'];
     $customer_city  = $_SESSION['city'];
     $customer_pcode =   $_SESSION['pcode'];
     $customer_number=   $_SESSION['number'];

     $sub_total=0;
     $shipping_cost = 0;
     $total = 0;
     if(isset($_POST['checkout'])){
       $fullname = $_POST['fullname'];
       $address  = $_POST['address'];
       $city     = $_POST['city'];
       $code     = $_POST['code'];
       $number   = $_POST['phone_number'];
       $invoice  = mt_rand();
       $date     = date("d-m-Y");

       $cartt = "SELECT * FROM cart WHERE cust_id='$customer_id'";
       $run  = mysqli_query($con,$cartt);
       if(mysqli_num_rows($run) > 0){
```

```
while($row = mysqli_fetch_array($run) ){
  $db_pro_id  = $row['product_id'];
  $db_pro_qty  = $row['quantity'];
$pr_query  = "SELECT * FROM gi_product WHERE pid=$db_pro_id";
$pr_run    = mysqli_query($con,$pr_query);
if(mysqli_num_rows($pr_run) > 0){
  while($pr_row = mysqli_fetch_array($pr_run)){

    $price = $pr_row['price'];
    $arrPrice = array($pr_row['price']);
    $single_pro_total_price = $db_pro_qty * $price;
    $checkout_query = "INSERT INTO `customer_order`(`customer_id`, `customer_email`,
    `customer_fullname`, `customer_address`, `customer_city`, `customer_pcode`, `customer_phonenumber`,
    `product_id`, `product_amount`, `invoice_no`, `products_qty`, `order_date`, `order_status`)
VALUES('$customer_id','$customer_email','$fullname','$address','$city','$code','$number',$db_pro_id,
    $single_pro_total_price,'$invoice',$db_pro_qty,'$date','pending')";
      if(mysqli_query($con,$checkout_query)){
        $del_query = "DELETE FROM cart where cust_id = $customer_id";
        if(mysqli_query($con,$del_query)){
         $_SESSION['message'] ="<div class='alert alert-primary alert-dismissible fade show pt-1 pb-1 pl-3' role='alert'>
          <strong><i class='fas fa-check-circle'></i> Thanks! </strong>for your order,It will be deliver within 7 working days.
          <button type='button' class='close p-2' data-dismiss='alert' aria-label='Close'>
            <span aria-hidden='true'>&times;</span>
          </button></div>";
          header('location:customer/orders.php');
        }
```

```
                }
               }
              }
             }
            }
        }
    ?>
        <h1>Check Out</h1>
        <div class="row">
          <!--shopping cart-->
         <div class="col-md-6 p-3">
         <h5>Shipping Detail</h5><hr>

         <div class="form-group">
          <label for="email"><b>Email:</b></label>
          <label><b><?php echo $customer_email;?></b></label>
         </div>

        <form method="post"  class="mt-4">
          <div class="form-group">
           <label for="fullname">Fullname:</label>
           <input type="text" name="fullname" placeholder="Full Name" class="form-control" value="<?php echo $customer_name; ?>" required>
          </div>

           <div class="form-group">
            <label for="address">Address:</label>
            <input type="text" name="address" placeholder="Address" value="<?php echo $customer_add; ?>" class="form-control" >
          </div>

          <div class="row">
           <div class="col-md-6 col-6">
             <div class="form-group">
```

```
                    <label for="city">City:</label>
                     <input type="text" name="city" placeholder="City" class="form-control" value="<?php echo $customer_city; ?>" required >
                    </div>
                   </div>

                   <div class="col-md-6 col-6">
                    <div class="form-group">
                    <label for="postalcode">Postal code:</label>
                     <input type="number" name="code" placeholder="Postal code" class="form-control" value="<?php echo $customer_pcode; ?>" required >
                   </div>
                   </div>

                  </div>

                  <div class="form-group">
                   <label for="number">Number:</label>
                   <input type="number" name="phone_number" placeholder="Phone Number" class="form-control" value="<?php echo $customer_number; ?>" required>
                  </div>

                  <div class="form-group text-center mt-4">
                     <input type="submit" name="checkout" class="btn btn-primary btn-block p-2" value="Place Order" id="border-less">
                  </div>
                 </form>
                </div>
               <!--end cart--->

               <!--shopping Detail-->
              <div class="col-md-6 p-3">
               <!-- cart-->
               <table class="table table-responsive table-hover ">
```

```
<h5>Order Detail</h5><hr>

 <tbody>
<?php
 $cart = "SELECT * FROM cart WHERE cust_id='$customer_id'";
 $run  = mysqli_query($con,$cart);
  if(mysqli_num_rows($run) > 0){
    while($cart_row = mysqli_fetch_array($run)){
       $db_cust_id = $cart_row['cust_id'];
       $db_pro_id  = $cart_row['product_id'];
       $db_pro_qty  = $cart_row['quantity'];

     $pr_query  = "SELECT * FROM gi_product WHERE pid=$db_pro_id";
     $pr_run    = mysqli_query($con,$pr_query);

      if(mysqli_num_rows($pr_run) > 0){
       while($pr_row = mysqli_fetch_array($pr_run)){
           $pid = $pr_row['pid'];
           $title = $pr_row['title'];
           $price = $pr_row['price'];
           $arrPrice = array($pr_row['price']);
           $state = $pr_row['state'];
           $img1 = $pr_row['image'];

           $single_pro_total_price = $db_pro_qty * $price;
           $pro_total_price = array($db_pro_qty * $price);
           $each_pr = implode($pro_total_price);
                    //   $values = array_sum($arrPrice);
             $shipping_cost=0;
             $values = array_sum($pro_total_price);
             $sub_total +=$values;
             $total = $sub_total + $shipping_cost;
```

```
                ?>
            <div class="row">
              <!--Image-->
               <div class="col-md-3 col-3">
                   <img src="img/<?php echo $img1;?>" width="100%">
               </div>
               <!--end image-->

                <!-- Title-->
               <div class="col-md-5 col-5">
                  <h5><?php echo $title;?> </h5>
                  <p> From:<?php echo $state;?></p>
               </div>
                  <!--end title-->

  <!--qunatity-->
<div class="col-md-2 col-1">
<h5>x <?php echo $db_pro_qty;?></h5>
</div>
<!--end qty-->
<!--price-->
<div class="col-md-2 col-2">
<h5>
<?php echo $single_pro_total_price;?>
</h5>
</div>
<!--end price-->

</div><hr>

<?php
```

```
            }
}
                    }
                    }
             ?>

              </tbody>
            </table>
          <!--end cart-->


          <div class="row">
            <div class="col-md-6 col-sm-6 col-6">
              <h6>Subtotal</h6>
              <h6>Shipping</h6>
              <h5 class="font-weight-bold">Total</h5>

            </div>
            <div class="col-md-6 col-sm-6 col-6">
              <h6 class="text-right font-weight-normal">INR <?php echo $sub_total;?></h6>
              <h6 class="text-right font-weight-normal">INR <?php echo $shipping_cost;?></h6>
              <h5 class="text-right font-weight-bold">INR <?php echo $total;?></h5>
            </div>
          </div>
        </div>
          <!--end order--->
      </div>
  <?php
 }
?>
  </div>
<?php include('include/footer.php');?>
```

SYSTEM TESTING & IMPLEMENTATION

System implementation is the important stage of the project when the theoretical designing is turn into practical system. The main stages in the implementation are as follows:

- Planning
- Training and
- Testing

Planning

When implementation is concerned planning plays an important role. The implementation of system will have to face any practical problems of controlling various activities of people outside their own data processing department. The active cooperation of different staff. Members will be essential for successful implementation.

Training

For successful implementation trained computer staffs are essential. So we have to select some staff and teach them about the computer implementation phase. Then only the designed system can be successfully implementation.

Testing

Before implementation the designed system should be tested with raw data to ensure that all modules of the system work correctly and satisfactorily. Software testing is a crucial element of software quality assurance and represents the ultimate review of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases. It was attempted to build software from an abstract concept to a tangible implementation. The testing phases responsible for ensuring that the product that has built performance the way that the detailed design documentation specific. The Lecturer information system was successfully tested with unit testing and system testing.

Unit Testing

At the very lowest level is unit testing where the programming that writes the codes as per the detailed specification. Unit testing comprises the set of testes performance by individual programmers' prior integration of the unite into a larger system.

This situation is illustrated as follow:

Coding & Debugging

- Unit test
- Integration test

System Testing

The next level of testing is the system testing where the project leader or system analyses texts all of the components to see that they are correct when combined as a system. A series of testing are performed for the newly developed system before the system is ready for user acceptance testing.

Validation Testing

Validation succeeds when the software function in a manner that can be reasonable expected by the customers. Software validation is achieved through a series of blank box text that demonstrate conformity with the requirements. Deviation or errors discovered at this step are corrected. Prior to the completion with the help of user by negotiating to established an evolving deficiency.

Output Testing

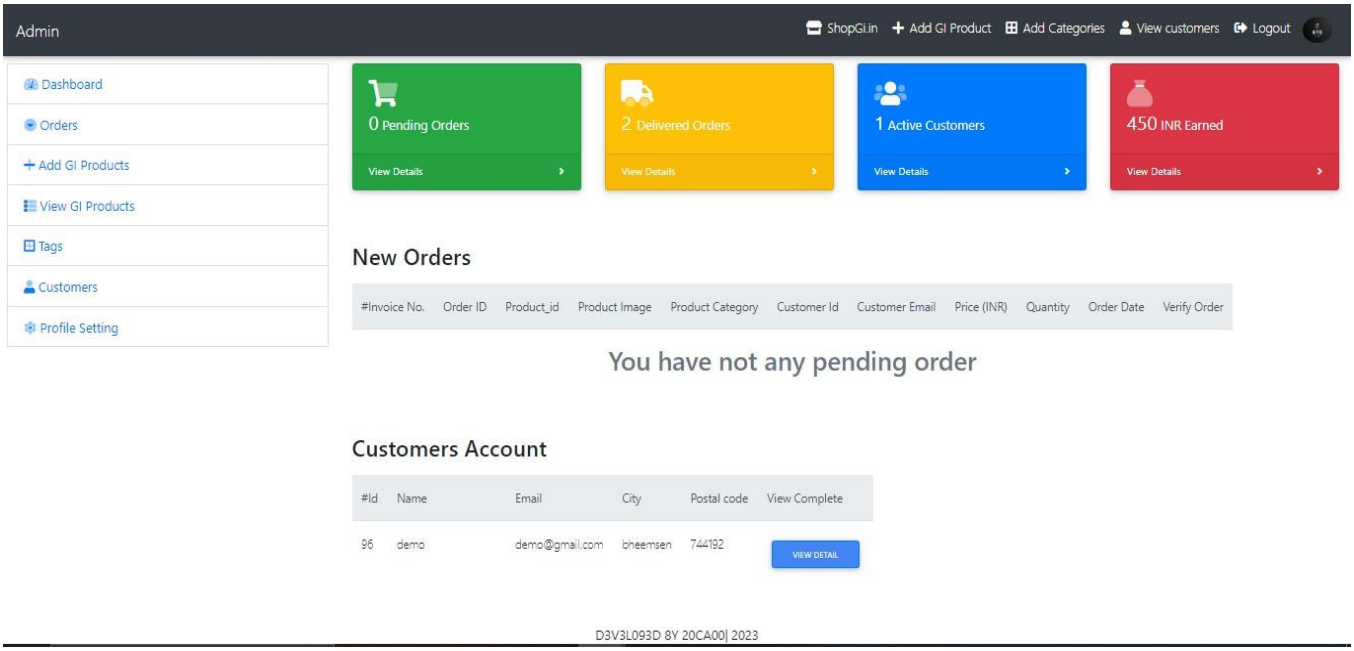
The output generated by the system under consideration is in the format required by the user. And the information in the reports are accurate. It is possible to generate timely reports without any errors.

User Acceptance Testing

The system developed is tested for acceptance by constantly keeping in touch with prospective system users at the time of development and changes were made whenever required. This is done with respect to the following input and output screen design online help message to guide the user.

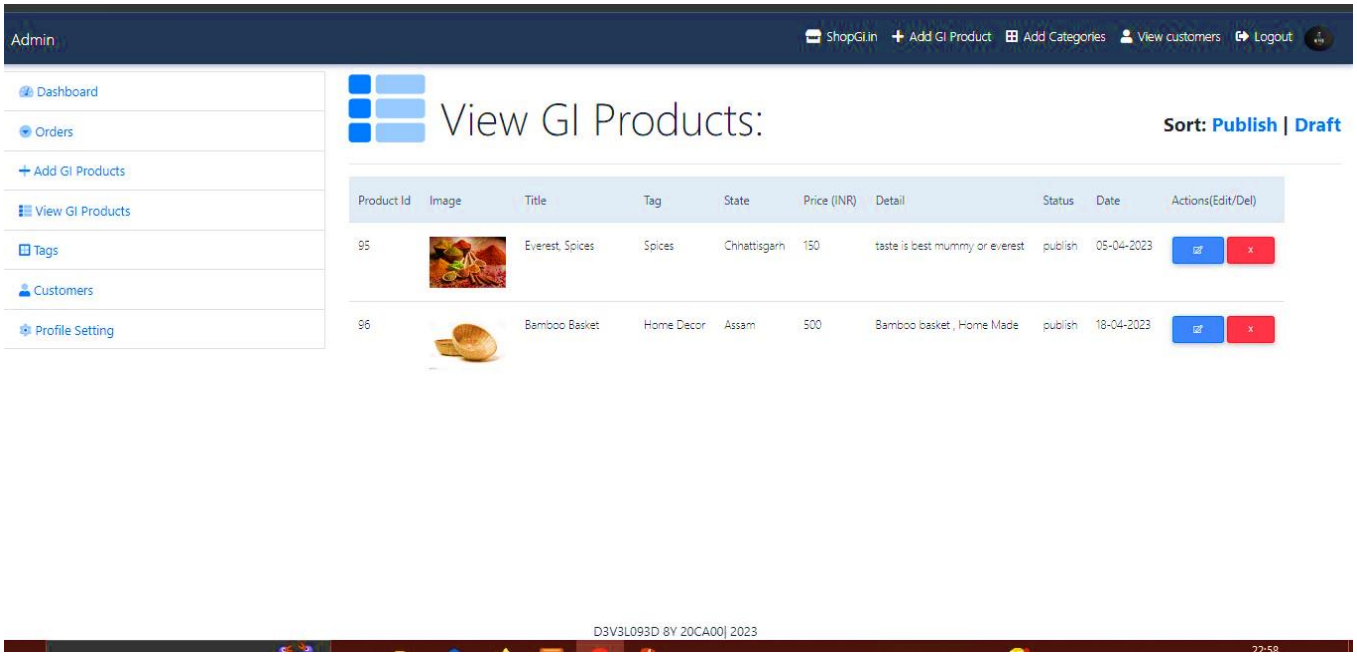
INPUT/OUTPUT SCREEN

- **Administration Dashboard**



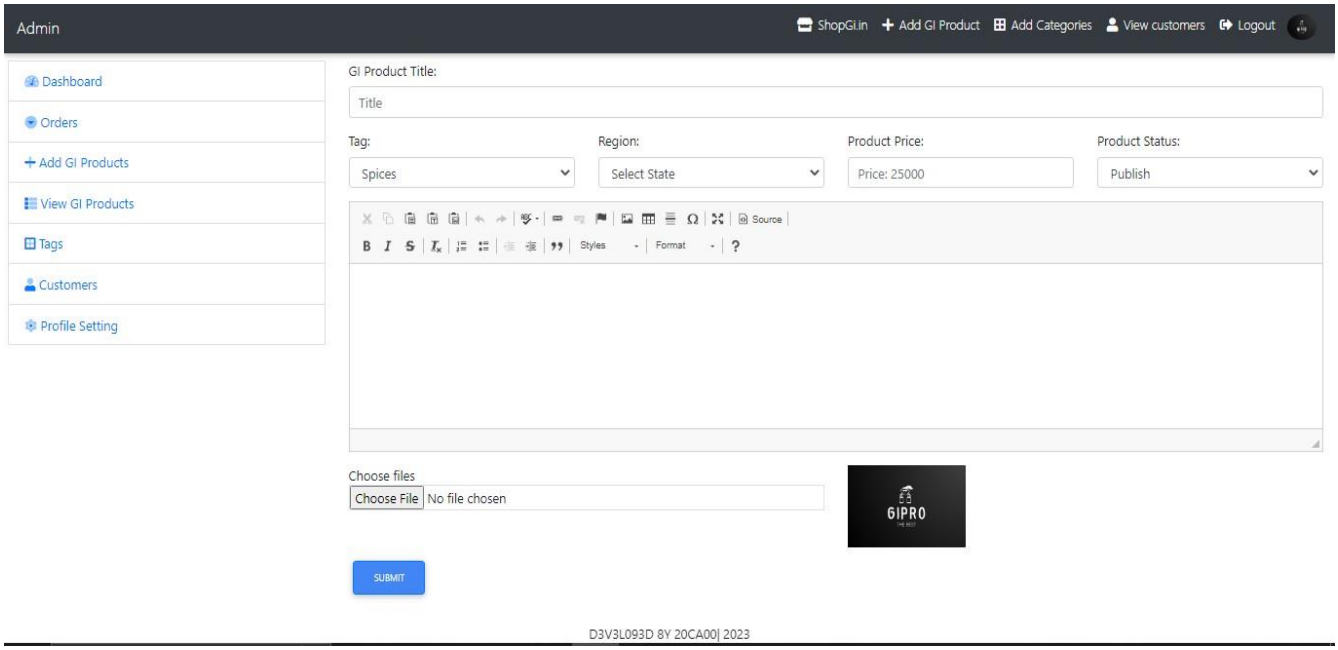
D3V3L093D 8Y 20CA00| 2023

- **Display Existing Products**

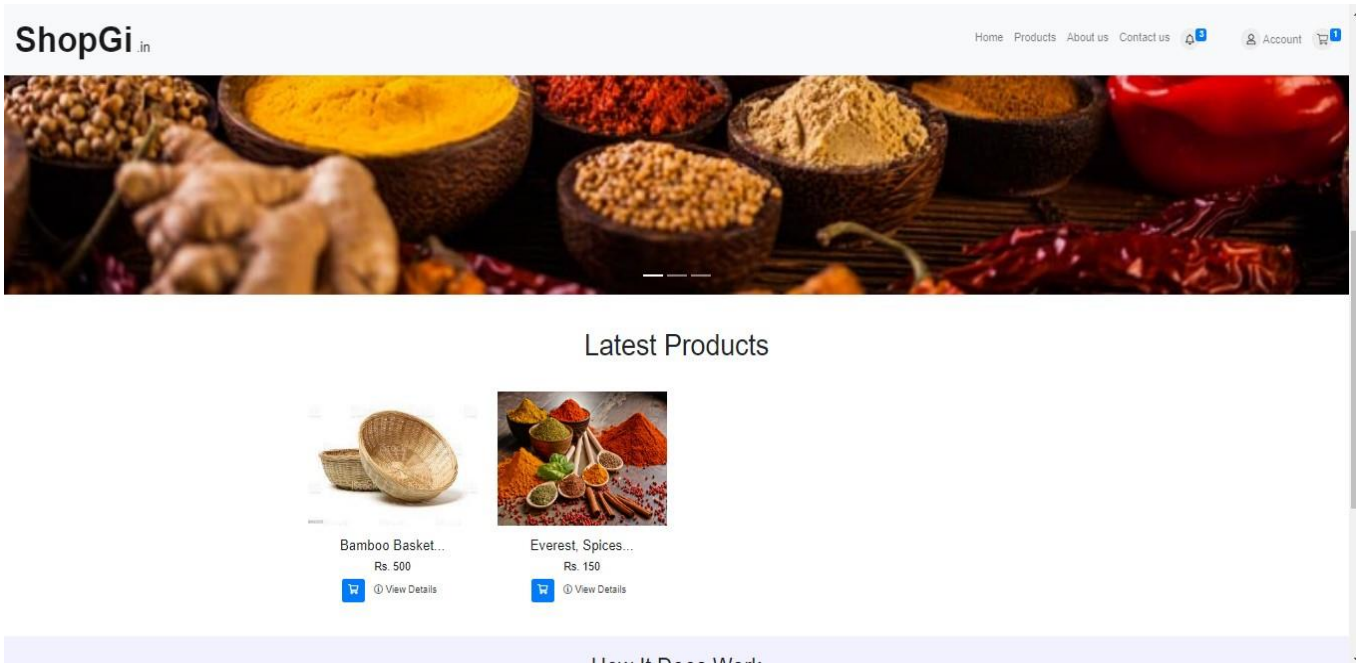


D3V3L093D 8Y 20CA00| 2023

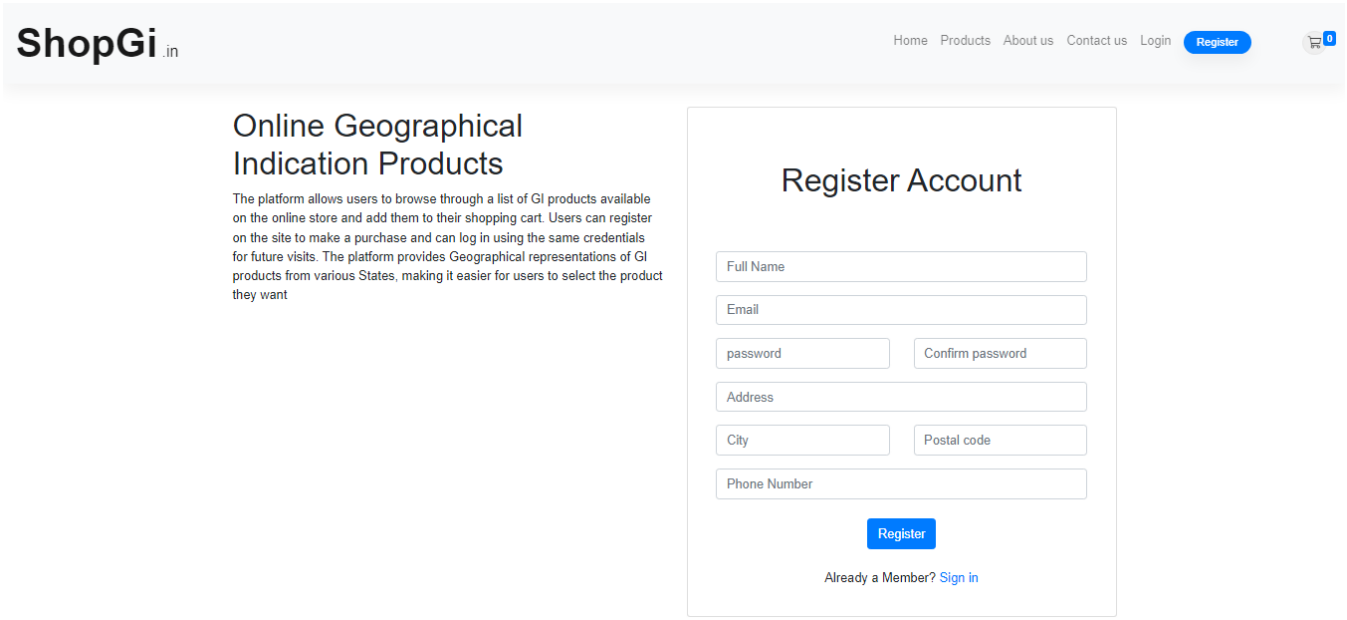
- **Add New Products**



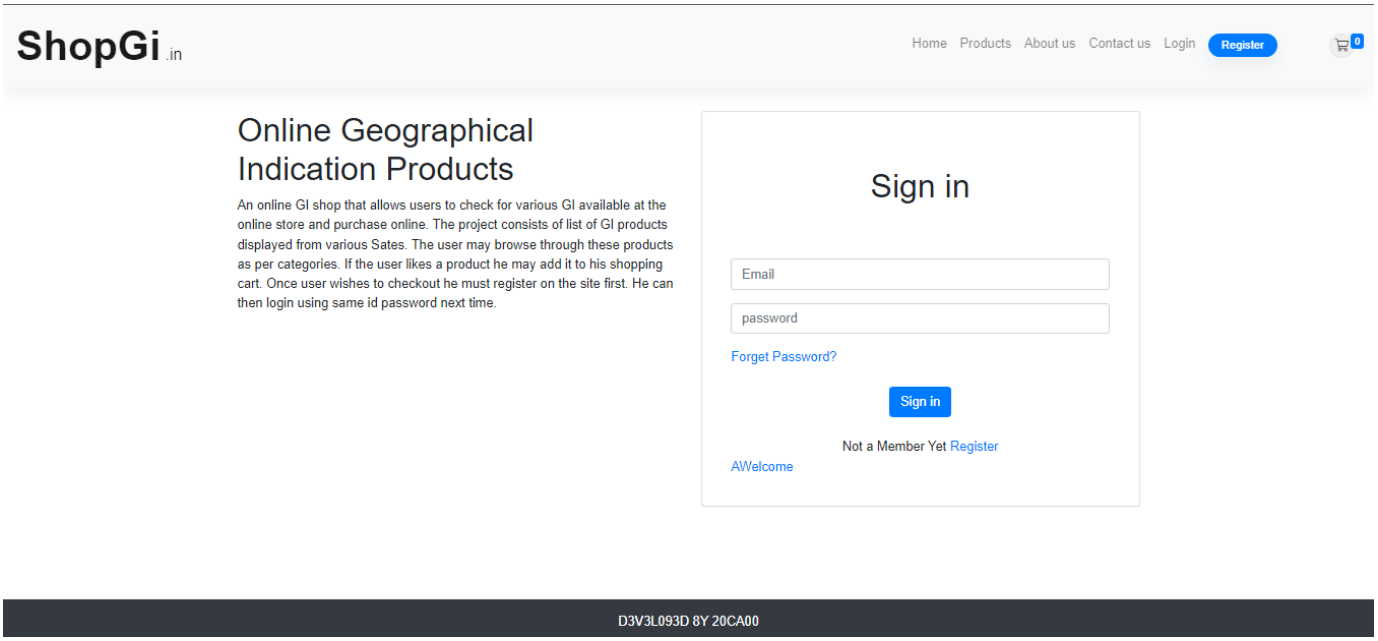
- **Display Products in Selling Dashboard**



- **User Registration**

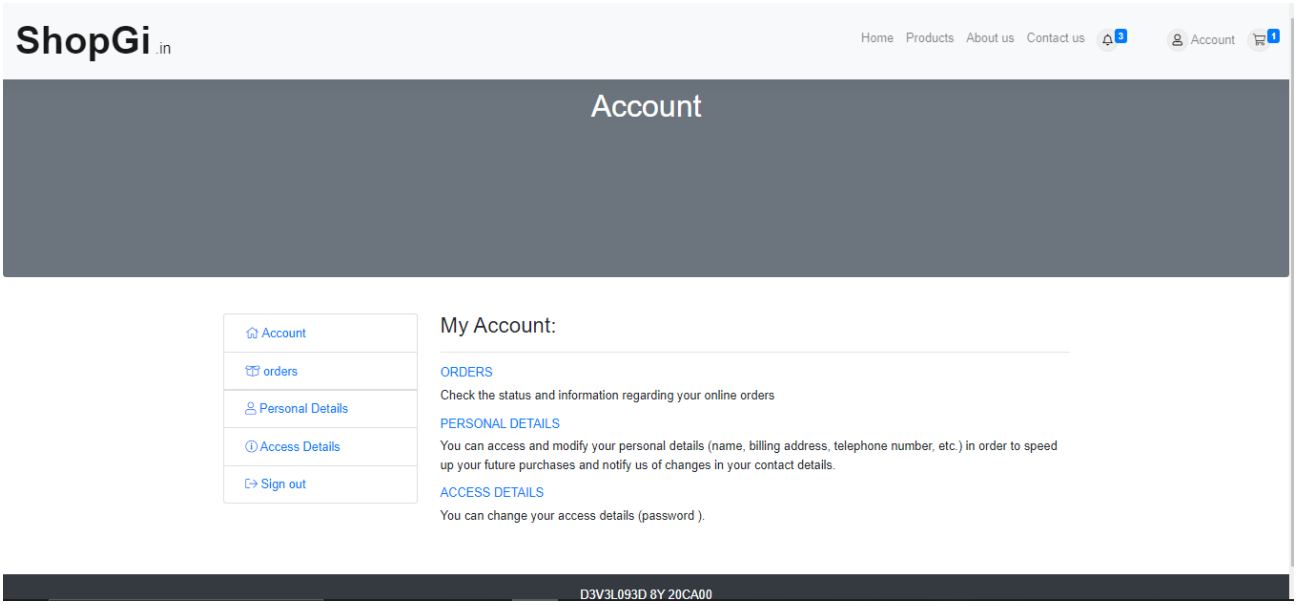


- **Customer Login**

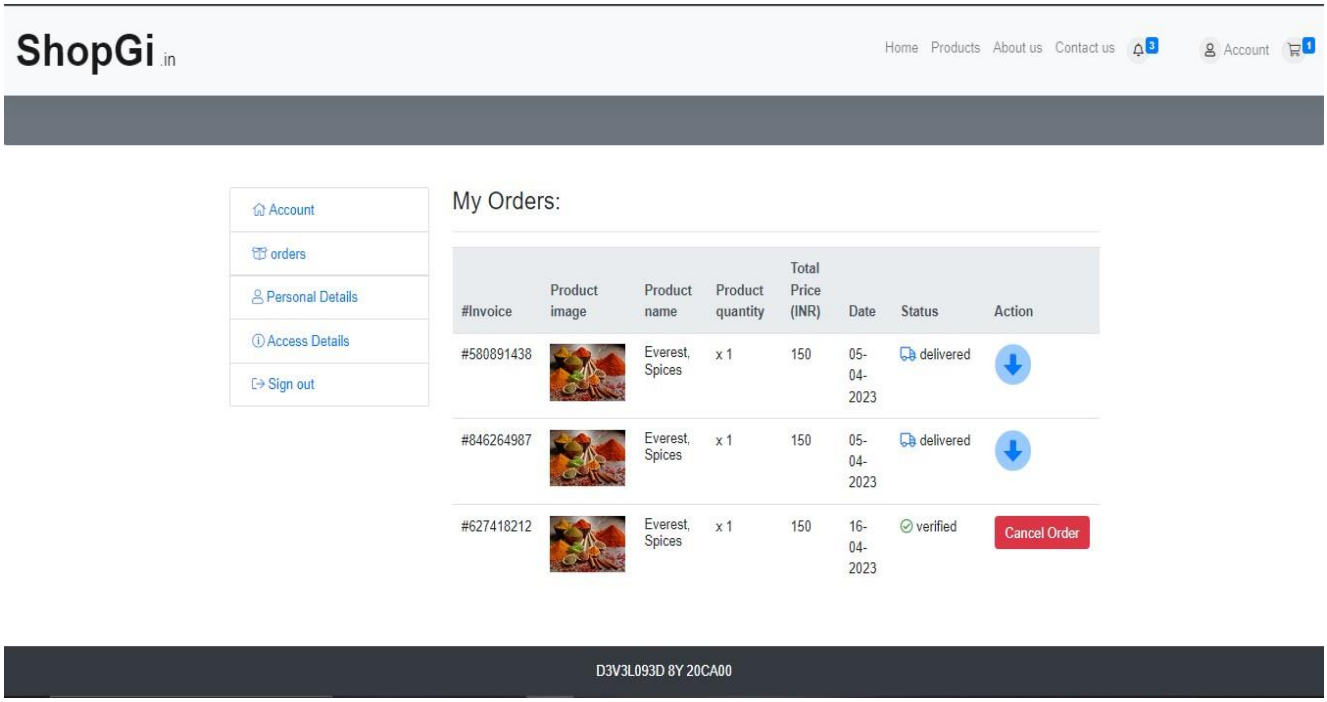


D3V3L093D 8Y 20CA00

- **Customer Dashboard**



- **Customer Orders**



➢ Invoice

ShopGi.in

JNRM, Port Blair , 744101
Phone +7430809922

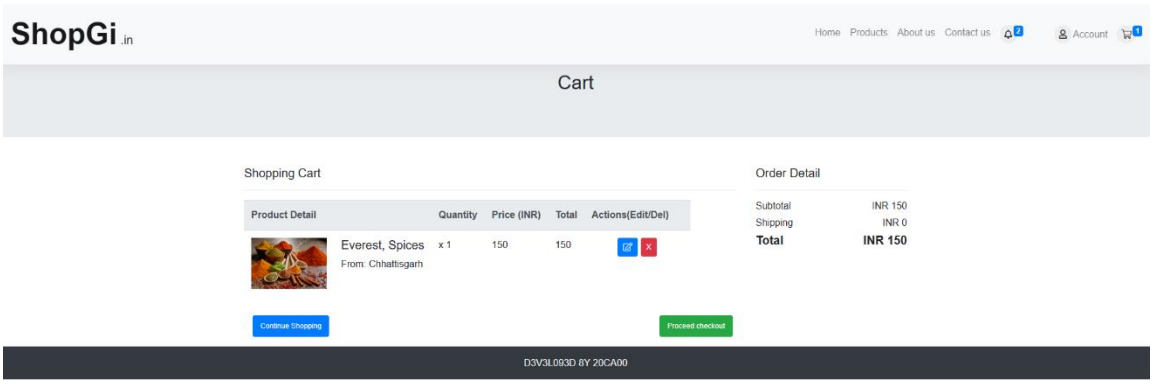
INVOICE

Date 05-04-2023
Invoice # 580891438
Customer ID 96

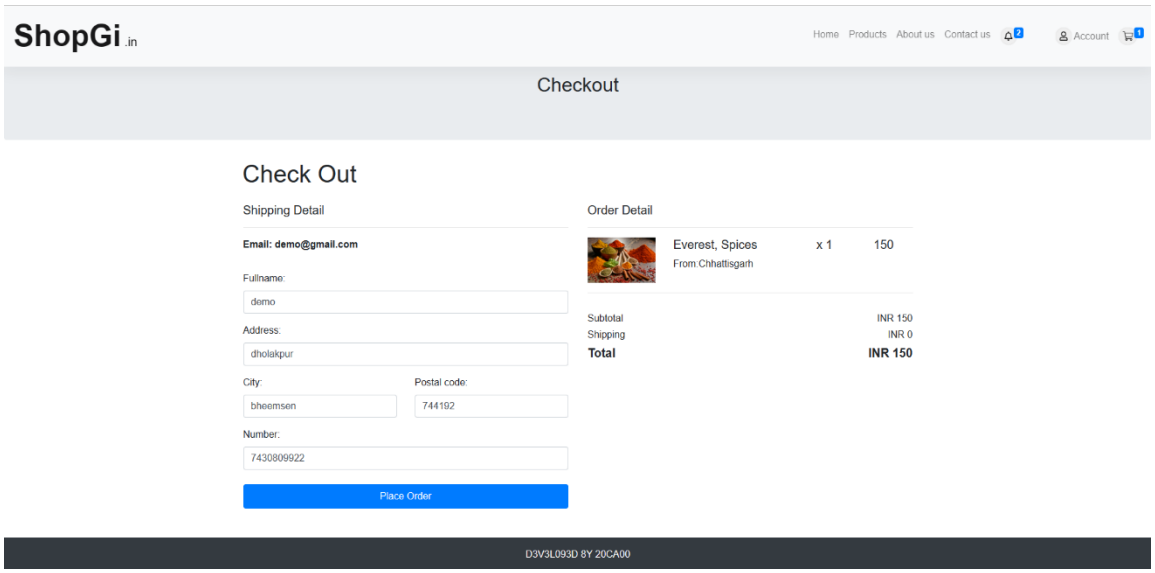
Bill to
demo
demo@gmail.com
dholakpur , bheemsen , 744192
7430809922

Description	Quantity	Single Amount	
Everest, Spices	1		150
	Shipping	Rs.	0
	Subtotal	Rs.	150
	Total Due	Rs.	150

➢ Customer Cart



➢ Order Place



LIMITATIONS OF THE PROJECT

Limitations for E Commerce Shopping Platform

There are several limitations for e-commerce shopping platforms, including:

Dependence on technology: E-commerce shopping platforms rely heavily on technology and internet connectivity. Any disruption or technical issue can result in significant downtime or loss of revenue.

Security risks: E-commerce platforms are vulnerable to cyber-attacks and data breaches, which can compromise customer data and affect the reputation of the platform. Lack of physical interaction: Customers cannot physically touch or examine products before purchase, which can result in misunderstandings and customer dissatisfaction. **Shipping and logistics:** E-commerce platforms need to rely on third-party shipping and logistics providers, which can lead to delays, lost or damaged products, and increased costs.

Customer service: Providing quality customer service can be challenging in e-commerce platforms, especially if the customers are spread across different regions or countries. **Competition:** E-commerce platforms face intense competition from other established players, making it challenging to attract and retain customers.

Legal and regulatory compliance: E-commerce platforms need to comply with various legal and regulatory requirements, such as data protection laws, consumer protection laws, and tax regulations, which can be complex and time-consuming.

FUTURE APPLICATIONS OF THE PROJECT

Improved product recommendations: Using machine learning algorithms, the platform can recommend products to customers based on their browsing and purchase history, as well as other relevant factors such as demographics and geographic location.

Personalized shopping experience: By analyzing customer data, such as browsing history, purchase history, and demographic information, the platform can provide a personalized shopping experience for each customer.

Streamlined checkout process: By integrating with payment gateways and optimizing the checkout process, the platform can improve the user experience and reduce cart abandonment rates.

Mobile app integration: By building a mobile app for the platform, customers can easily shop on-the-go, receive push notifications for promotions and discounts, and have access to their order history.

Voice-enabled shopping: With the increasing popularity of voice assistants like Alexa and Google Assistant, integrating voice-enabled shopping into the platform could provide a convenient and innovative shopping experience for customers.

BIBLIOGRAPHY

- https://W3Scholl.com
- https://GitHub.com
- https://Stackoverflow.com
- https://Greecksofgreeks.com
- https://Tutorialspoint.com
- https://youtube.com
- PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide
- PHP & MySQL – Novice to Ninja