PROJECT PROFILE

Project Title	Shopsphere Ecommerce Website
Project Description	This is an e-commerce website project named ShopSphere developed using PHP and MySQL. The homepage displays different product categories like Men's Clothing, Women's Wear, Accessories, and Footwear, each with attractive images and links to explore collections. The product section showcases items such as makeup kits, cameras, watches, and laptops with prices and an Add to Cart option. A search bar is also included for finding products easily. The website features a clean pink-themed design with footer details like services, social links, quick links, and contact information. Overall, it provides a simple yet effective shopping platform interface for users.
Front End	HTML, CSS, JavaScript
Back End	PHP
Other Tools	MS WORD
Guide	Prof. Ripal Pandya, Prof. Milind Anandpara
Submitted To	BKNM University,Junagadh(S.S.S.D.I.I.T College)

USE OF SYSTEM DEVELOPMENT LIFE CYCLE MODEL

Software Development Life Cycle (SDLC) is a process for development of software. There are some steps to follow to create a software application.

In an SDLC the steps follow requirement gathering. In requirement gathering questionnaire, personal interview etc. are the method for gathering information. Analysis phase includes creating Software Requirement Specification and analyze the gathered data. In design phase, design of Software application i.e. database design and GUI design have to be prepared. In coding phase, coding is done of different modules and forms. In testing phase, the different type of testing is done like integration testing, unit testing, system testing and at last the created software is implemented and maintained.

- **Following are the different Life Cycle Model example:**

  - **Waterfall model**
  - **Iterative waterfall model**
  - **Prototyping model**
  - **Evolutionary**
  - **R.A.D. model (Rapid Application Development)**

FEASIBILITY STUDY

The main aim of feasibility study is to determine whether developing the project is functionally and technically feasible or not.

The feasibility study involves analysis of the problem and collection of data which world be input to the system, the processing required to be carried out on these data, the output data required to be product by the system, as well as study of various constraint on the behavior of the system.

An initial determine in a proposal that whether an alternative system is feasible or not. To determine feasibility of candidate system in all respect I need to consider following feasibility factors:

There three types of feasibility study:

1) Technical

2) Operational

3) Economical

1. Introduction

The project "ShopSphere" is an e-commerce web application designed to provide users with an online platform to browse, search, and purchase products such as electronics, fashion items, accessories, and cosmetics. It includes features like product listings, shopping cart functionality, search options, and category-wise browsing.

2. Technical Feasibility

- **Platform: Developed using PHP (as seen in URL: index.php) with HTML, CSS for frontend.**
- **Database: Likely MySQL or similar for storing product details, user accounts, and transactions.**
- **Hosting Requirements:**

->Localhost/XAMPP for development.

->Requires cloud hosting or shared hosting for deployment.

- **Skills Required:**

->PHP, MySQL, HTML, CSS, JavaScript.

- **Feasibility:** **Technically feasible as all required technologies are widely available and manageable.**

3. Economic Feasibility

- **Cost Estimation:**

->Development tools (XAMPP, VS Code, PHPMyAdmin) – Free.

->Hosting & Domain – ~$50–100 annually.

->Maintenance – depends on scale.

- **Revenue Generation:**

->Product sales commission.

->Advertisements or featured products.

- **Feasibility:** **Low-cost development and high market demand make it economically viable.**

4. Operational Feasibility

- **Users: Customers (buyers), Admin (manages products).**
- **Operations Supported:**

  -> Product browsing by category.

  ->Adding items to cart.

  ->Searching products.

  ->Secure login/registration.

- **Ease of Use: Simple UI with category browsing and search functionality.**
- **Feasibility: Operationally practical for small to medium-scale use.**

5. Legal Feasibility

- **Must ensure compliance with e-commerce regulations such as data privacy (GDPR/Indian IT Act).**
- **Copyright of product images must be respected.**
- **Feasibility: Legally feasible if guidelines are followed.**

6. Scheduling Feasibility

- **Estimated Development Timeline:**

-> Frontend design: 2–3 weeks.

-> Backend (database & PHP): 3–4 weeks.

-> Testing & deployment: 2 weeks.

-> Total Duration: ~2 months for a basic functional site.

- **Feasibility: Achievable within academic/project deadlines.**

REQUIREMENT GATHERING

Questionnaire:

1) What do you want to create a software or a website?

➢ I want create a website.

2) What does your business do?

➢ Our business provides services for all types of men's Tshirt and Jeans, Makeups, Laptop, Mobile ect products.

3) What type of title do you want?

➢ I want a title like "ShopSphere Ecommerce Website".

4) How Many modules do you want?

➢ Two modules: Admin and User (Client).

5) What is the time period allocated for this project?

➢ 92 Days (3 Month)

6) What features do you need in your website?

➢ I want features like Add Data, Update Data, Delete Data and duplicate can not allowed.

7) What is the budget allocated for the web development project?

➢ No specific budget has been decided yet.

8) Do you need a map location of your shop on the contact page?

➢ Yes, but the shop address should be included.

9) Do you need online payment system?

➢ No.

REQUIREMENT ANALYSIS

The requirement gathering phase of the Software Development Life Cycle (SDLC) involves acquiring information from the institution for which the project is being developed. This process is essential to understand the user's requirements and the specific functionalities they expect from the system. Several techniques are used for information gathering, including:

- **Questionnaire: To understand user preferences, shopping behavior, and expectations.**
- **Observation: Analyzing existing e-commerce platforms to identify common features and improvements.**
- **Personal Interviews: Conducting discussions with potential users (customers and admins) to determine their needs and preferred functionalities.**
- **Record Review: Studying existing grocery store operations to optimize online management.**

-> Main Requirements of the System:

- **For Customers (Users):**

1. Browsing and Discovery :

-> Easy Navigation: The customer needs to be able to easily find what they are looking for. The website should have a clear navigation bar with links to key sections like "Home," "Products," "Contact," "Register," and "Login."

-> Product Categories: The customer wants to browse products by specific categories. The "Shop By Category" section on the home page with options like "Men's Clothing," "Women's Wear," "Accessories," and "Footwear" fulfills this need.

-> Search Functionality: A search bar is essential for customers who know exactly what they want. They should be able to type in keywords (e.g., "watch," "laptop") and get relevant results quickly.

-> Product Details: When browsing products, the customer expects to see a clear product image, the product name (e.g., "camera," "watch"), a brief description, and the price. The ability to "Add to cart" directly from this view is a key feature.

2. User Account Management :

-> Simple Registration: A straightforward registration form is necessary for a new customer to create an account. The required information should be minimal

and clear: name, email, phone number, address, password, and gender.

-> Easy Login: The customer needs a simple login form to access their account. This form should only require an email and password.

-> Data Security: Customers need assurance that their personal information, especially their password, is handled securely. The website should protect this data.

3. User Interface (UI) and Experience (UX) :

-> Clean and Organized Design: The website should have a visually appealing and organized layout. The use of a consistent color scheme (pink and blue) and clear sections (e.g., footer with social links, contact info) contributes to a good user experience.

-> Intuitive Layout: The placement of the search bar, product categories, and "Add to cart" buttons should be logical and easy to understand for first-time visitors.

-> Contact Information: The customer needs a way to contact the business. The footer provides essential contact details, including a location and phone number.

4. Post-Purchase Requirements (Inferred) :

-> While not explicitly shown in the images, a customer will expect a seamless experience after adding items to the cart. This includes:

-> Shopping Cart: A way to view, edit, and manage items in their cart.

-> Secure Checkout: A secure process to enter payment and shipping information.

-> Order Confirmation: An email or on-screen message confirming their order.

-> Order Tracking: A way to track the status of their order.

• For Admin:

1.Dashboard :

->The dashboard serves as the central hub, providing a quick overview of key metrics.

-> Requirement: Display high-level statistics in a clear, summarized format.

-> Metrics to display:

- **Total Users**
- **Total Products**
- **Total Orders**

- **Total Delivered Orders**

-> User Interface: Use colored boxes or cards to highlight each metric for easy visibility.

2. User Management :

-> The "Users" section allows the admin to view and manage registered users.

-> Requirement: Display a list of all users in a tabular format.

-> User details to display:

- **User Name**
- **Email**
- **Phone**
- **Address**

-> User Interface: A table with clear column headers. There are no "Delete" or "Update" buttons visible, suggesting this might be a read-only view for the admin, but a more robust system might require these functionalities.

3. Product Management :

-> This is a critical section with multiple sub-functions:

a. Add Products :

->Requirement: Provide a form for adding new products to the store.

->Form fields required:

- **Title**
- **Description**
- **Price**
- **Quantity**
- **Product Image**

->Action: A "Add product" button to submit the form data and save the new product.

b. View Products :

->Requirement: Display a comprehensive list of all products in a table.

->Product details to display:

- **Title**
- **Description**
- **Quantity**
- **Pric**
- **Image**

-> Admin actions:

->Delete: A "Delete" button for each product row to remove it from the database.

->Update: An "Update" button for each product row to navigate to the product update page.

c. Update Product :

-> Requirement: Provide a pre-filled form to modify an existing product's details.

->Form fields required:

- **Title**
- **Description**
- **Pric**
- **Quantity**
- **Current Image**
- **Change Image**

-> Action: An "Update product" button to submit the modified data. The URL (update_product.php?id=7) indicates that the product ID is passed as a parameter for specific updates.

4. Order Management : The "Orders" section is crucial for managing the fulfillment process.

-> Requirement: Display a list of all orders placed by customers.

-> Order details to display:

- **Customer Name**
- **Email**
- **Address**
- **Phone**
- **Product Title**
- **Price**
- **Image**
- **Status (e.g., "Delivered")**

Admin actions:

->Change Status: A button or control to update the status of an order. The image shows a "Delivered" button, suggesting this might be a one-click action to change the status to "Delivered" once the order has been shipped and received. The system may require more granular statuses (e.g., "Pending," "Processing," "Shipped," "Delivered").

1) Hardware and Software Requirements :

• Hardware Requirements: • A computer with 4 GB RAM or higher (8 GB recommended for better performance)

• A minimum of 256 GB SSD storage or higher.

• An Intel Core i3 processor or higher.

• Software Requirements:

• Visual Studio Code.

• SQL Server for database management.

2) Front-End Tools :

• HTML, CSS, JavaScript(for structuring and styling web pages)

3) Back-End Tools :

• php programming language for application logic.

• SQL Server for data storage and management.

4) Other Tools & Technology Used :

• Apache And XAMPP.

Project Abstracts (User Roles & Capabilities)

User Roles:

1. Admin

- **Capabilities:**
  - **View Dashboard: The dashboard provides an overview of key metrics, including:**
    - **Total number of users .**
    - **Total number of products.**
    - **Total number of orders.**
    - **Total delivered orders.**
  - **Manage Users: The admin can view a list of all users registered on the site, along with their details like name, email, phone number, and address.**
  - **Manage Products: The admin has full control over the product catalog. This includes:**
  - **Adding Products: An admin can add new products by providing a title, description, price, quantity, and product image.**

- **Viewing Products: The admin can view a list of all products, showing their title, description, quantity, price, and image.**
- **Updating Products: An admin can edit existing product details, such as the title, description, price, quantity, and image.**
- **Deleting Products: An admin can remove products from the catalog.**
- **Manage Orders: The admin can view all orders placed on the site. For each order, they can see details such as the customer's name, email, address, phone number, product title, price, and the current status (e.g., "Delivered"). The "Change Status" button suggests they can also update an order's status, likely to reflect its progress from "processing" to "shipped" or "delivered."**

2. User (Customer)

• Capabilities:

- **View Products: Guests can view all the products listed on the website, including their images, descriptions, and prices.**
- **Search for Products: They can use the search bar to find specific items.**
- **Browse Categories: Guests can browse products by category, such as "Men's**

Clothing," "Women's Wear," "Accessories," and "Footwear."

- **Access Public Pages: They can navigate to public pages like "Home," "Products," and "Contact" as seen in the navigation bar.**
- **Register an Account: The "Register" link allows guests to create a new user account by filling out the registration form with their name, email, phone, address, password, and gender.**
- **Log in to an Account: The "Login" link allows them to log in to an existing account using their email and password.**
- **Add to Cart: They can add products to a shopping cart, which is a key e-commerce function for making a purchase.**
- **Manage Account: Although not explicitly shown in the images, a registered user would typically have the ability to view and edit their profile information, check their order history, and manage their address book. This is implied by the presence of a "Login" and "Register" system.**
- **Checkout and Purchase: The ultimate capability of a registered user is to proceed with the checkout process and complete a purchase.**

PROPOSED SYSTEM

-> Role of the Website :

The website is an e-commerce platform named "ShopSphere." It functions as an online store where users can browse and purchase products. The images show a user-facing front end and an administrative back end. The site's purpose is to facilitate the online buying and selling of goods.

-> User-facing role: Allows customers to view products by category (men's clothing, women's wear, accessories, footwear) and search for specific items. It has features for user registration, login, and adding products to a cart.

-> Administrative role: Provides a backend dashboard for managing the store. The admin can view and manage users, add new products, update or delete existing products, and view and track all customer orders.

1) Technology Stack :

- **Backend: The URLs consistently show .php extensions (e.g., index.php, register.php, adminpage.php), which indicates the use of PHP as the server-side scripting language. This is a**

common choice for web development, especially for dynamic sites.

- **Frontend: The styling and layout suggest the use of standard web technologies:**
  - **HTML: For the structure of the web pages.**
  - **CSS: For styling, including the use of colors (pink and blue palettes), layout, and the design of forms and product cards.**
- **Database: The presence of user, product, and order management implies the use of a database to store this information. A common database for a PHP application is MySQL.**
- **Server: The URLs begin with localhost, which means the website is running on a local development server. This is typically achieved using a solution like XAMPP or WAMP which bundles Apache (the web server), MySQL, and PHP.**

2) Development Environment :

- **Localhost: The website is currently hosted on a local server, as shown by the localhost in the browser's address bar. This is a standard practice for development and testing before deployment to a live server.**

- **Operating System: The taskbar at the bottom of the screen shows the Windows 10/11 interface, indicating that the development is being done on a Windows machine.**
- **Code Editor: The taskbar icon that looks like the Visual Studio Code logo suggests the developer is likely using VS Code as their primary code editor.**

3) System Advantages :

- **Centralized Management: The admin panel provides a single, easy-to-use interface to manage all aspects of the e-commerce business, from product inventory to customer data and order fulfillment. This streamlines operations and reduces manual effort.**
- **Scalable Architecture: The use of a LAMP stack (Linux, Apache, MySQL, PHP) or WAMP/XAMPP provides a solid, well-documented, and scalable foundation. This architecture can handle a growing number of products and users.**
- **User-Friendly Interface: The website's front end is clean and visually organized with clear sections for categories, products, and a simple login/registration process. This enhances the user experience, making it easy for customers to find what they need.**

- **Customization and Flexibility: Since the site is built on PHP and uses a standard database, it can be easily customized and extended with new features as the business grows.**

- **Accessibility: The website can be accessed from any device with a web browser, providing a wide reach to potential customers.**

ADVANTAGES & LIMITATIONS OF PROPOSED SYSTEM

Advantages:

- **Centralized Management: The admin dashboard acts as a single control panel, allowing the owner to efficiently oversee all aspects of the business, from inventory to sales.**
- **User Data Organization: The "All Users" page provides a clear and organized list of registered customers, including their names, emails, phone numbers, and addresses, which is crucial for customer relationship management.**
- **Comprehensive Product Control: The system empowers the admin to easily add new products, update details of existing ones, and remove them from the store, ensuring the inventory is always up-to-date.**
- **Clear Order Tracking: The "All orders" section offers a transparent view of all sales, showing who bought what and the current delivery status. This simplifies the fulfillment process and reduces errors.**
- **Simplified User Experience: The front-end is designed for easy navigation, featuring a search bar and clear product categories, which helps customers find items quickly and intuitively.**

- **Core E-commerce Functionality: The system successfully implements essential features like user registration, login, and the ability to view products and add them to a cart (implied by the "Add to cart" button).**

Limitations:

- **No Advanced Search or Filtering: The products page includes a basic search bar but lacks advanced filtering options, such as the ability to sort by price, category, or customer rating.**
- **Limited Payment Options: There is no visible information about payment gateway integrations, which suggests the system may not support online payments and might be limited to a manual or "cash on delivery" process.**
- **Lack of Customer Features: The system does not appear to have a user account or profile page where customers can view their order history, save addresses, or manage their personal information after registration.**
- **No Product Review or Rating System: The images do not show any functionality for customers to leave reviews, ratings, or feedback on products. This limits social proof and trust-building for new buyers.**

PERT CHART AND GANTT CHART

Gantt Chart:

Gantt charts are mainly used for scheduling, budgeting, and resource planning. It allocates resources to activities including staff, hardware, software, etc.

A Gantt chart is a special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of time planned for the corresponding activity.

A Gantt chart is part of a PERT chart, and we can get many things from the Gantt chart via the PERT chart. The Gantt chart describes time scheduling and resource planning. When more than one activity is done in parallel, this chart is very useful for understanding each process.

A Gantt chart is a special type of bar chart where each bar describes the time allocated for the activity. In a Gantt chart, each bar consists of task parts and schedule parts. The schedule part indicates the length of time estimated for the process.

So we can say that the Gantt chart is the most important part of a report because it describes total planning and estimation with the date of our project.

	01-07-25 To 01-08-25	01-08-25 To 05-08-25	05-08-25 To 10-08-25	10-08-25 To 20-08-25	20-08-25 To 31-08-25	31-08-25 To 10-09-25
Requirement	✓					
Feasibility study		✓				
Data Dictionary			✓			
Designing				✓		
Developing					✓	
Testing						✓

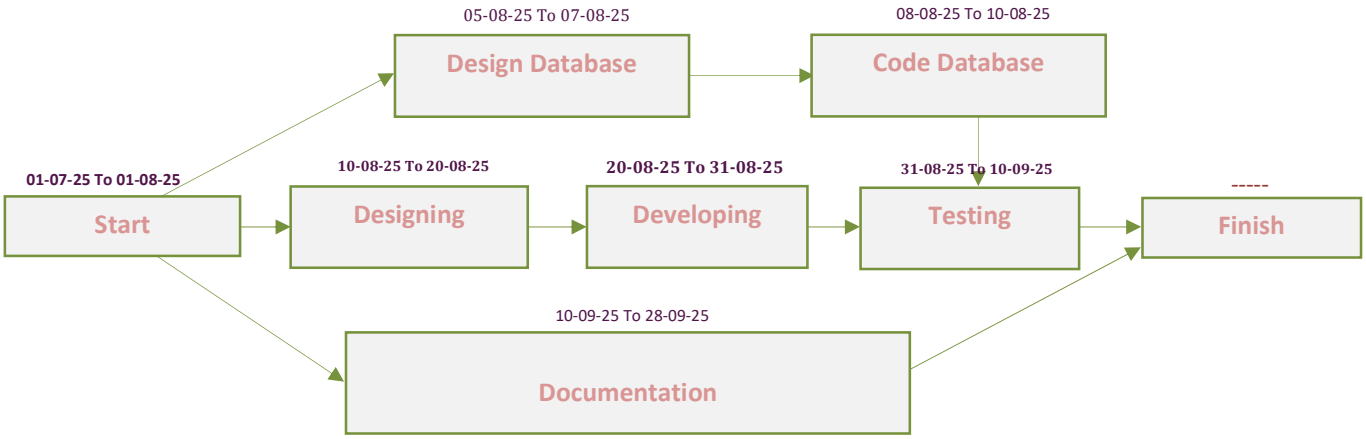
PERT Chart:

PERT chart is very use full to show the project statistically PERT chart is also very use full in the project assuming.

It also monitoring on the activities by the time. It task the care of the process by timing.

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies. PERT charts are a more sophisticated form of activity chart. Where instead of making a single estimate for each task, pessimistic, likely and optimistic estimates are made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. There are thus not one but many critical paths, depending on the permutations of the estimates for each task. This makes analysis of critical path show by using shaded boxes.

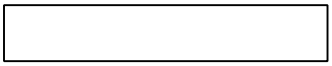
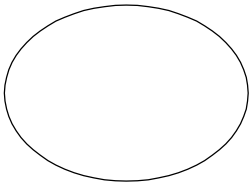
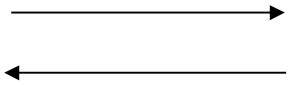
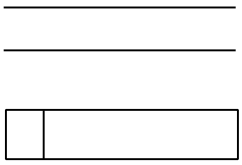
Gantt chart can be derived automatically from PERT charts. However, PERT charts cannot be automatically derived from Gantt charts because PERT charts incorporate additional information about the time when an engineer doses a task. This information is not available is helpful in planning the utilization of resources, while the PERT charts is more useful for monitoring the timely progress of activities.


05-08-25 To 07-08-25
Design Database
08-08-25 To 10-08-25
Code Database
01-07-25 To 01-08-25
Start
10-08-25 To 20-08-25
Designing
20-08-25 To 31-08-25
Developing
31-08-25 To 10-09-25
Testing

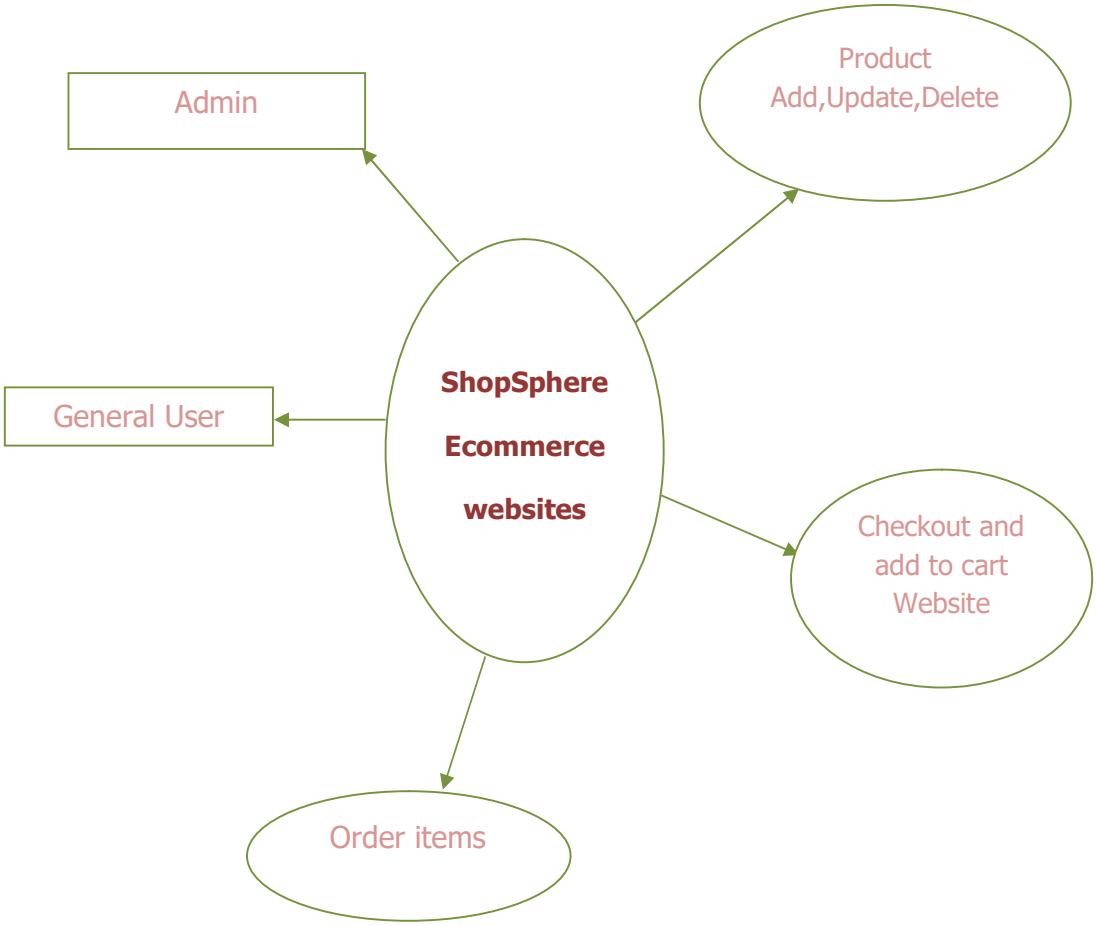
Finish
10-09-25 To 28-09-25
Documentation

DATA FLOW DIAGRAM

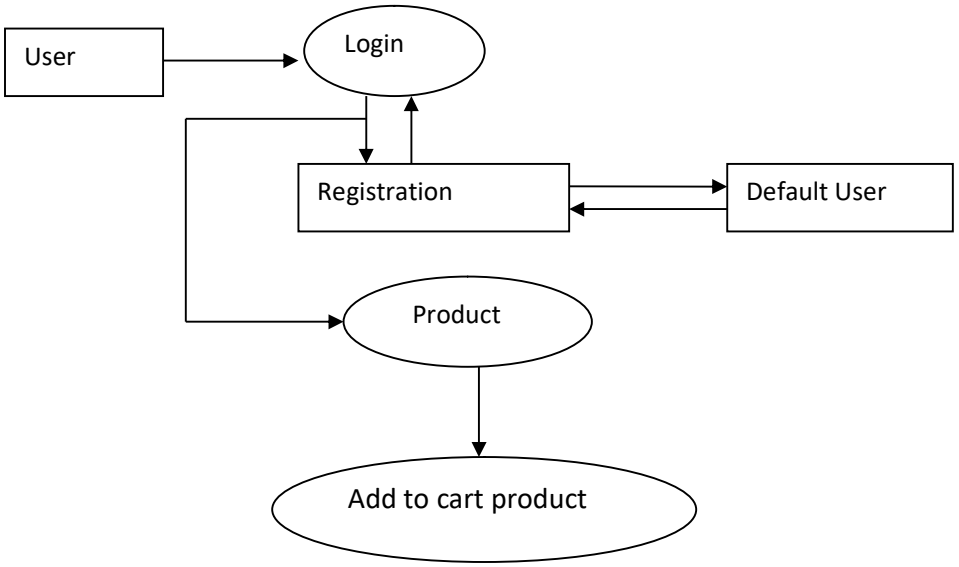
For Understanding:

Symbol	Name	Use
	External Entity	Rectangle source and / sink destination data.
	Process / Function	Transformed, Store, or Distribute. Annotated with number and name of function.
	Data Flow	Direction of data flow single piece of data or logical collection of data.
	Data Store	Open Rectangle Parallel lines Data Structure, File, Table, Databases.

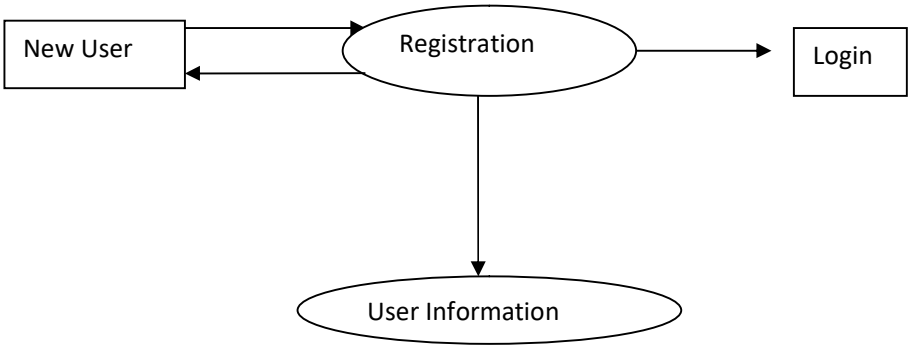
Context Level Diagram (0 Level Diagram)



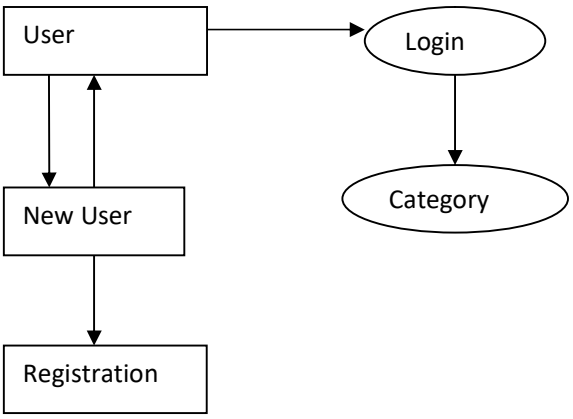
1st Level Diagram (user)



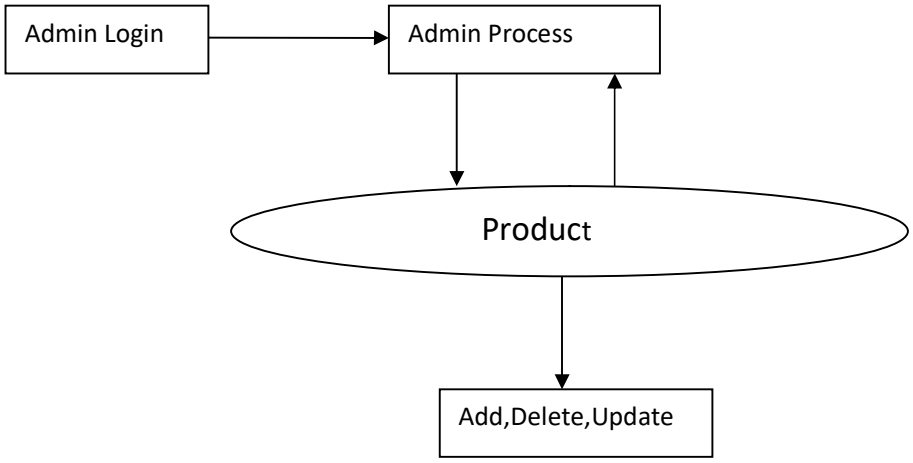
➔ Registration Process :



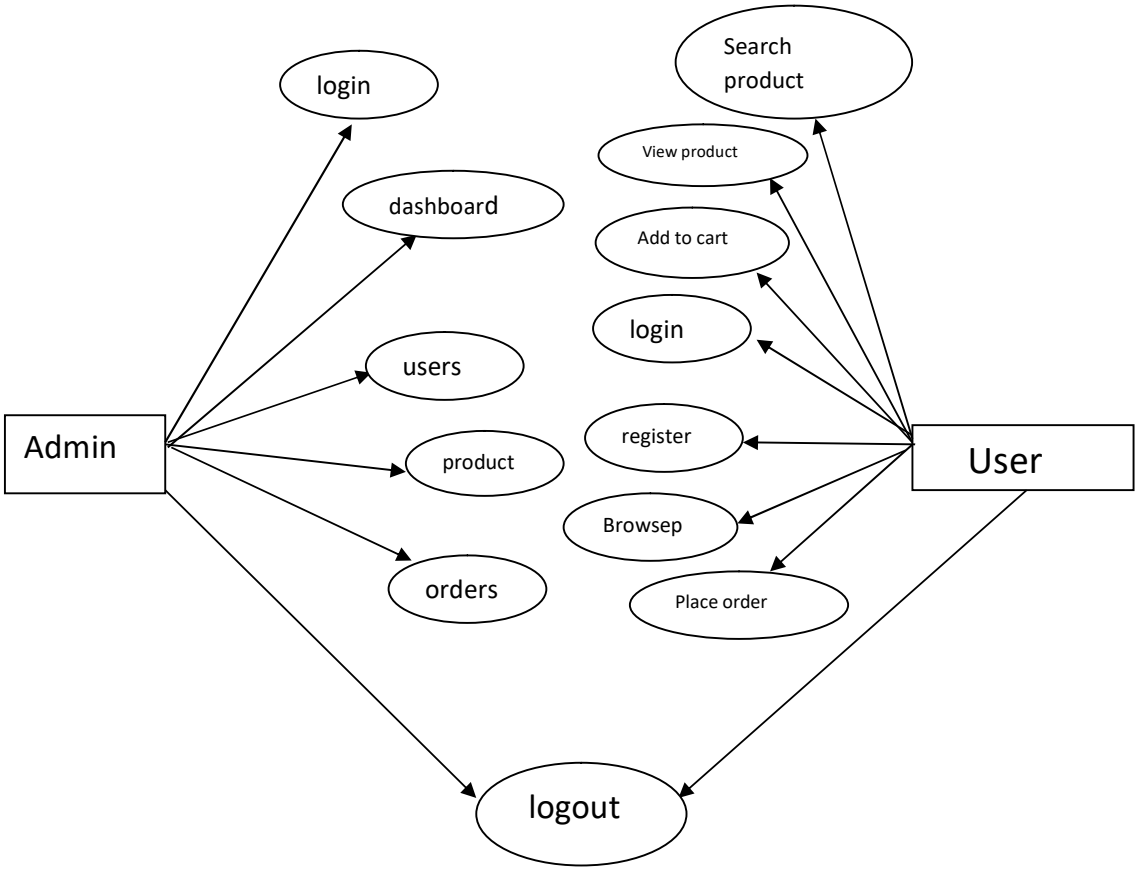
➔ Login Process



2nd Level Diagram (Admin)

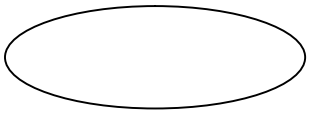


USE CASE DIAGRAM



A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagrams the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

Acter



Use Case

An actors is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Actor and use case descriptions show the detailed interaction between the actor and their use case. The descriptions enable a proper understanding of how actors interact.

FLOW CHART

A Flow is a pictorial representation of an algorithm. Programmers often use it as a program-planning tool for visually organizing a sequence of steps necessary to solve a problem using computer. It uses boxes of different shapes to denote different type of instructions. The actual instructions are written within these boxes using clear and concise statements. Solid lines having arrow marks connect these boxes to indicate the flow of operation, that is, the exact sequence in which to execute the instructions. The process of drawing a flowchart for an algorithm is known as flowcharting.

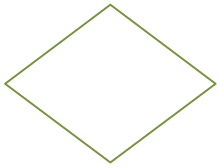
Basic Flowchart Symbols:

Terminal

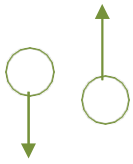
Input/Output

Flow Lines

Processing

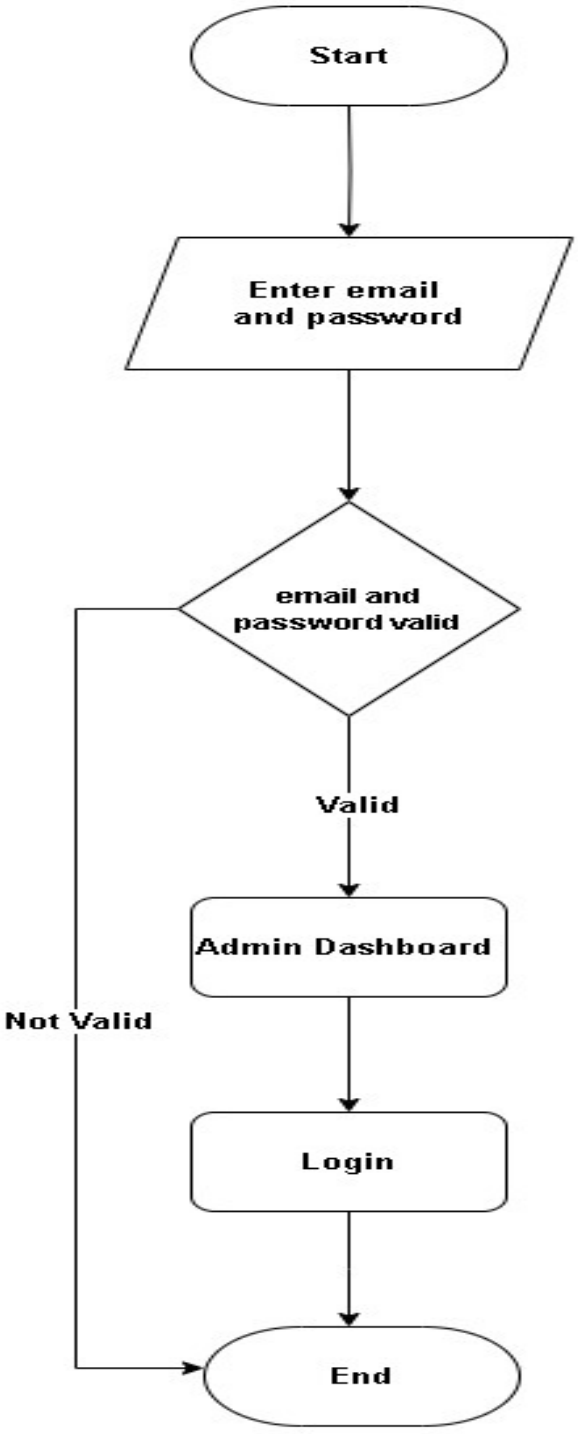


Decision

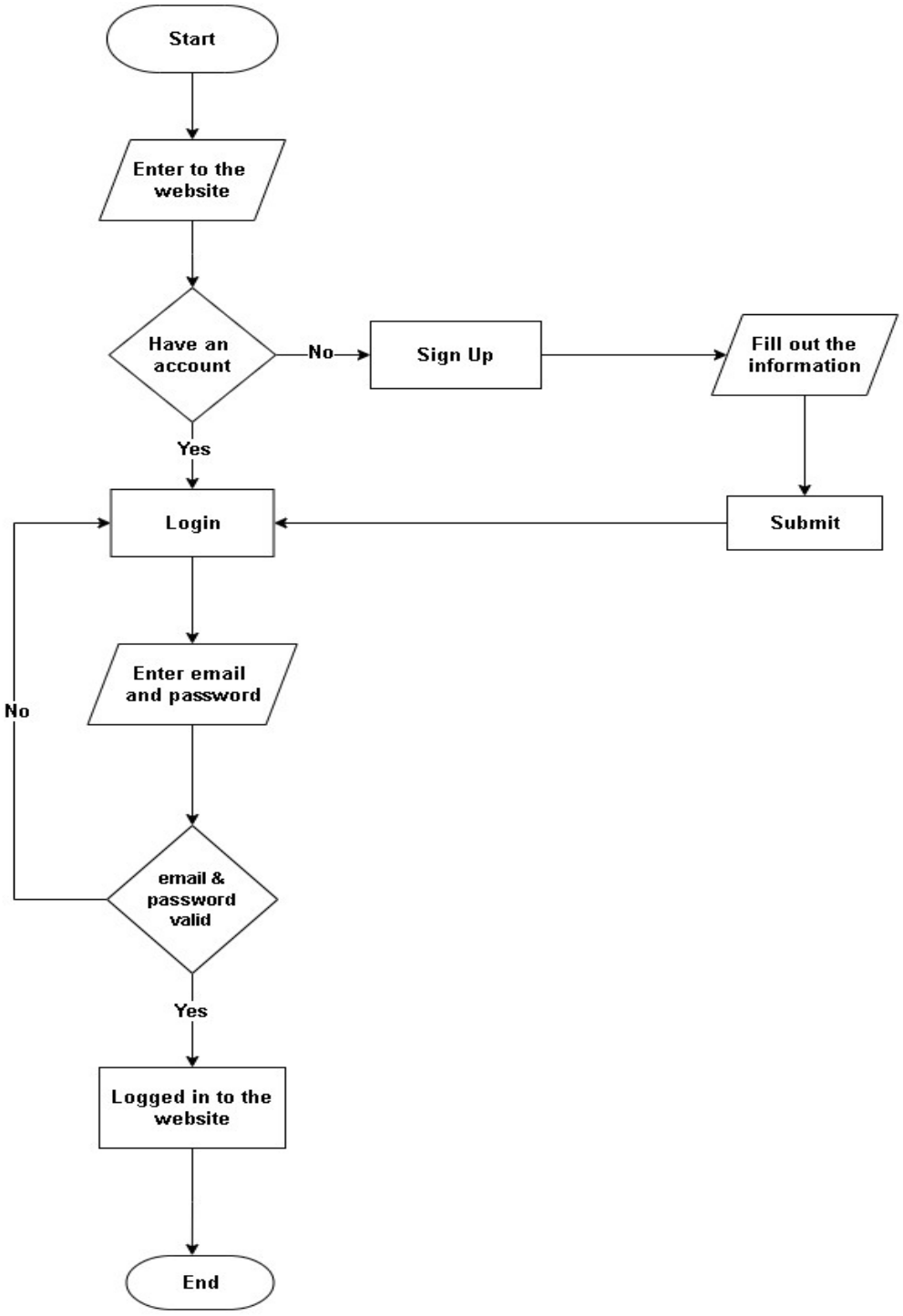


Connectors

Admin Login Flowchart



User Login Flowchart



COST ESTIMATION

Working Time Estimation

- Duration: 3 months and 2 days • Days per Month: 30 days
- Total Days:

3months×30days/month+2days=92days

- Working Hours per Days: 3.5 hours
- Total Working Hours:

92days × 3.5hours/day=322hours

Cost Breakdown

Description	Cost (₹)
Computer Rent	5,000
Light Bill (320 units X ₹7/unit)	2,240
Database Design & Creation	2,000
Coding & Validation	6,000
Total	**15,240**

DATA DICTIONARY & NORMALIZATION

Data dictionary set of information describing the contents format, and structure of a database and the relationship between its elements, used to contact of access to and manipulate of the database.

Database Name: php_ecom

Table 1: orders

Column Name	DataType (Size)	Constraints
Id	Int(50)	A.I, P.K
Title	Varchar(50)	UNIQUE, NOT NULL
Price	Varchar(50)	NOT NULL
Image	Varchar(255)	NULL
Username	Varchar(50)	NOT NULL
Email	Varchar(50)	UNIQUE, NOT NULL
Phone	Varchar(50)	UNIQUE, NOT NULL
Address	Varchar(50)	NOT NULL
Status	Varchar(50)	DEFAULT 'Pending', NOT NULL

Table 2: products

Column Name	DataType (Size)	Constraints
Id	int(50)	A.I, P.K
Description	longtext	NOT NULL
Price	Varchar(50)	NOT NULL
Quantity	Varchar(50)	DEFAULT 1, NOT NULL
Image	Varchar(255)	NULL
Title	Varchar(50)	UNIQUE, NOT NULL

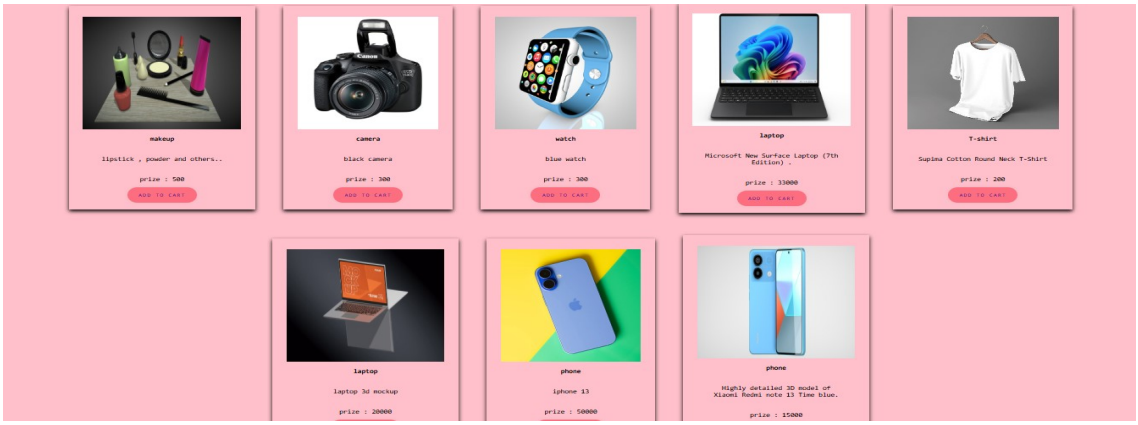
Table 3: users

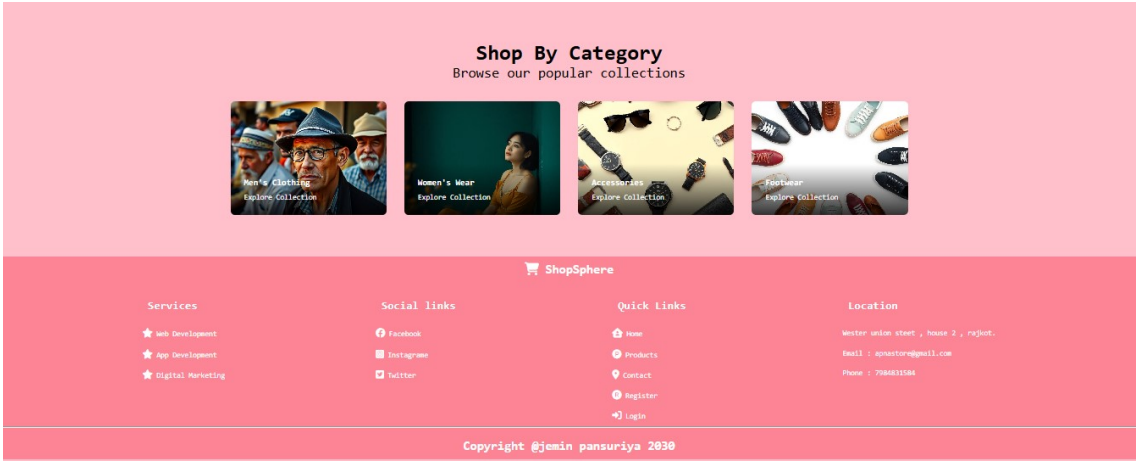
Column Name	DataType (Size)	Constraints
Id	int(50)	A.I, P.K
Name	Varchar(50)	NOT NULL
Email	Varchar(50)	UNIQUE, NOT NULL
Password	Varchar(50)	NOT NULL
Phone	Varchar(50)	UNIQUE, NOT NULL
Address	Varchar(50)	NOT NULL
User type	Varchar(50)	NOT NULL
Gender	enum('m', 'f')	NOT NULL

SCREEN LAYOUTS

Client side:

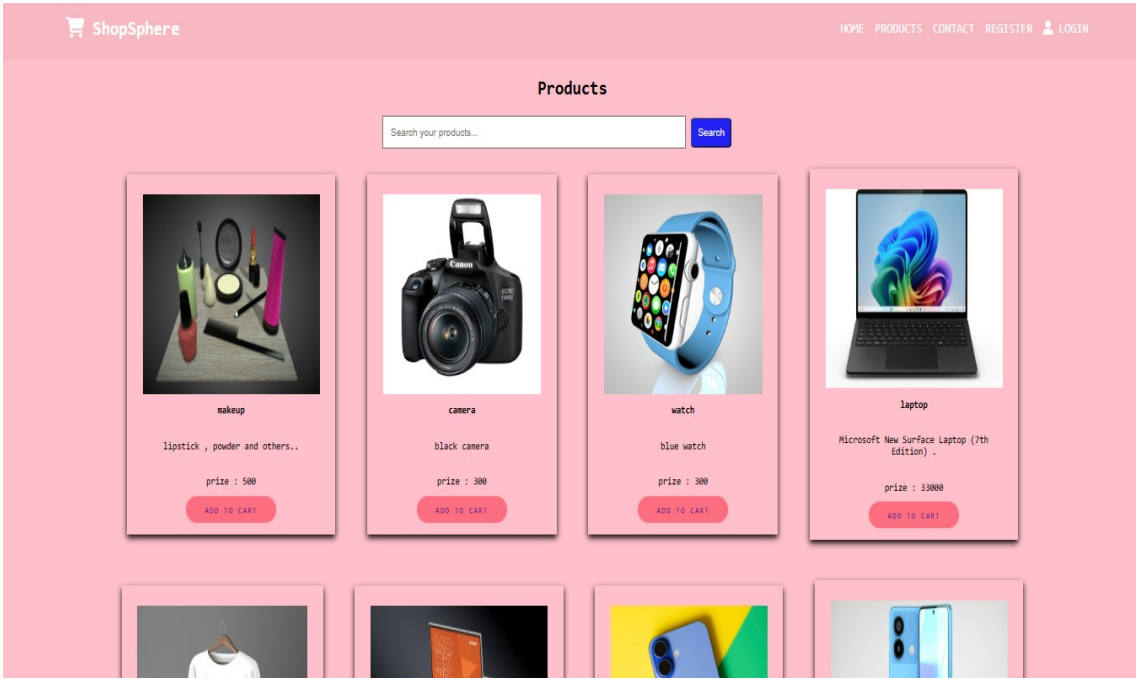
Home page:





```
<?php
session_start();
error_reporting(0);
$conn = mysqli_connect("localhost","root","","php_ecom");
if(isset($_GET['search']))
{
  $search_value = $_GET['my_search'];
  $sql = "SELECT * from products where concat(title,description) LIKE '%$search_value%'";
  $result = mysqli_query($conn,$sql);
}
else{
$sql = "SELECT * from products";
$result = mysqli_query($conn,$sql);
}
?>
```

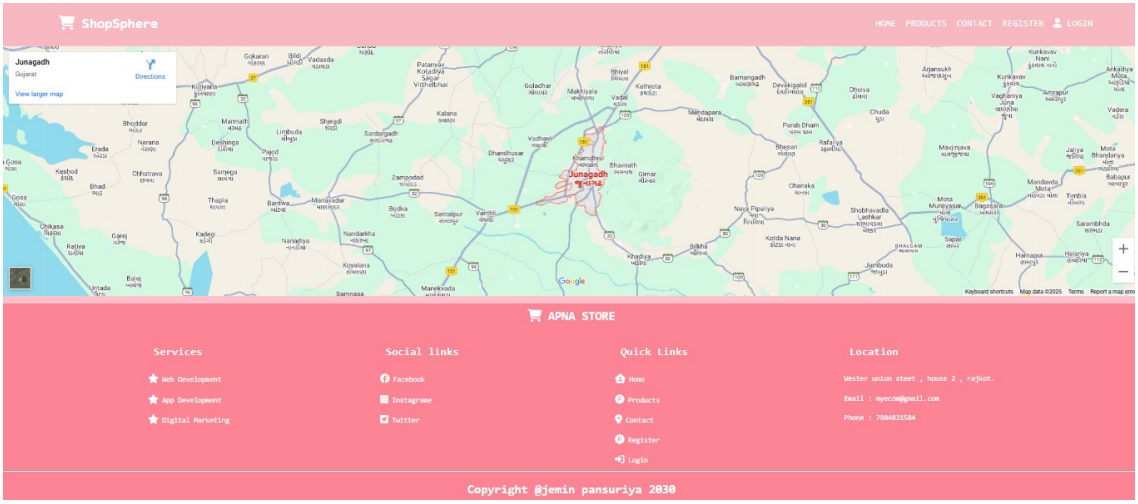
Product page :




```
<div class="my_card">
```

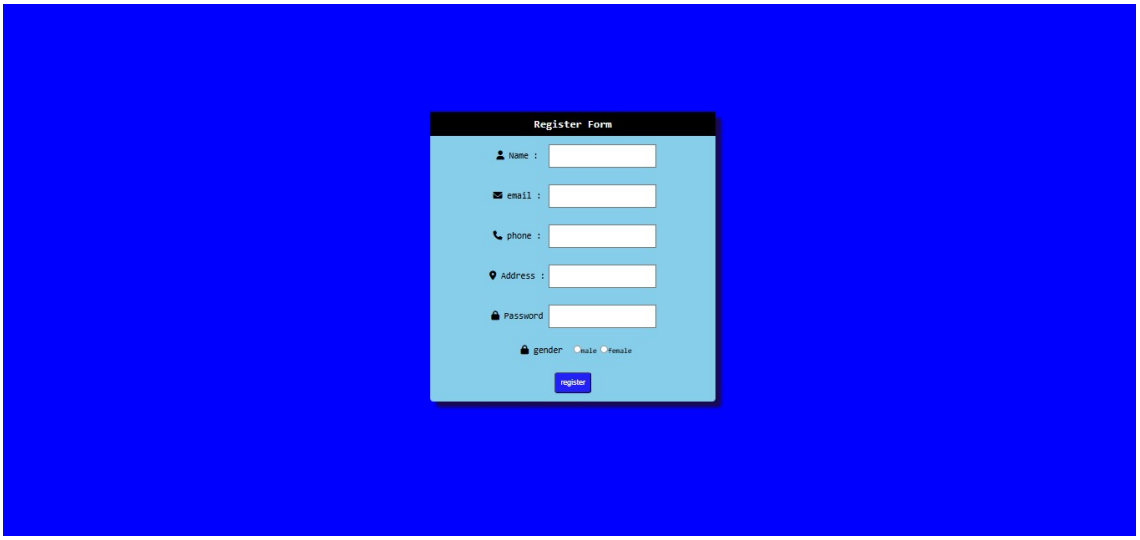
```
<?php
 while($row=mysqli_fetch_assoc($result))
 {
 ?>
<div class ="card">
<img class="p_image" src="product_image/<?php echo $row['image'] ?>">
<h4><?php echo $row['title'] ?></h4>
<p><?php echo $row['description'] ?></p>
<p>prize : <?php echo $row['price'] ?></p>
<?php
if($_SESSION['user_email'])
{
  ?>
  <a href="my_order.php?id=<?php echo $row['id'] ?>&email=<?php echo
$_SESSION['user_email'] ?>">add to cart</a>
  <?php
}
else
{
  ?>
  <a href="home/login.php">add to cart</a>
<?php
}
 ?>
 </div>
 <?php
 }
?>
</div>
```

Contect page:



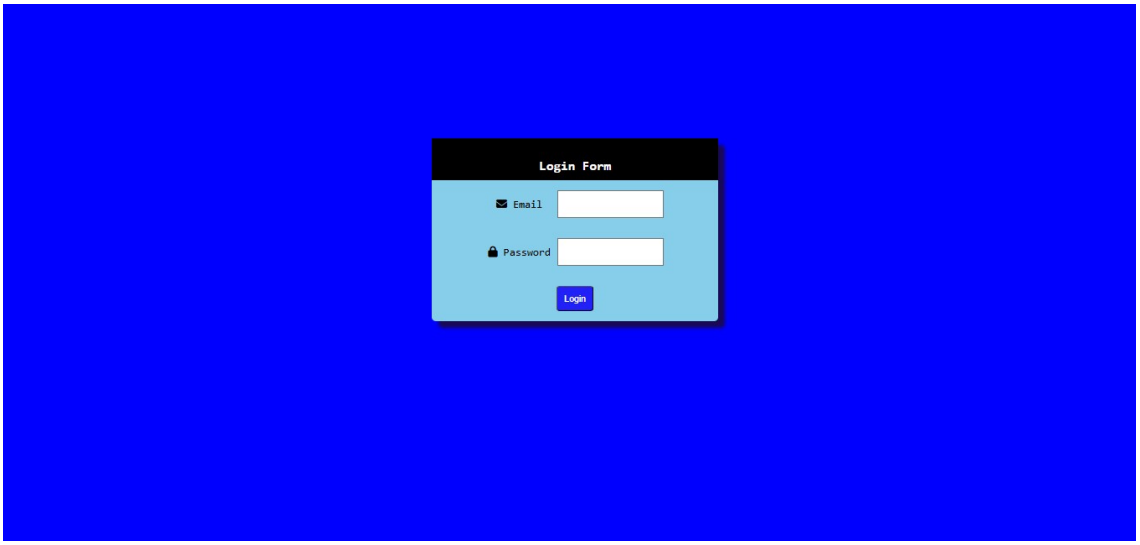

```
<?php
session_start();
error_reporting(0);
$conn = mysqli_connect("localhost","root","","php_ecom");
if(isset($_GET['search']))
{
   $search_value = $_GET['my_search'];
   $sql = "SELECT * from products where concat(title,description) LIKE '%$search_value%'";
   $result = mysqli_query($conn,$sql);
}
else{
$sql = "SELECT * from products";
$result = mysqli_query($conn,$sql);
}
?>
```

Register page :




```
<?php

$conn = mysqli_connect("localhost","root","","php_ecom");

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

   $u_name = $_POST['name'];

   $u_email = $_POST['email'];

   $u_address = $_POST['address'];

   $u_phone = $_POST['phone'];

   $u_pass = $_POST['password'];

   $usertype = "user";

   $u_gen = $_POST['gender'];

   $sql = "INSERT INTO users (name,email,password,phone,address,usertype,gender) VALUES
('$u_name','$u_email','$u_pass','$u_phone','$u_address','$usertype','$u_gen')";

   $data = mysqli_query($conn,$sql);

   if($data)

   {

     echo  "Register success";

   }

}

?>
```

Login page(admin,clint) :




```
<?php
session_start();
error_reporting(0);
$conn = mysqli_connect("localhost","root","","php_ecom");
if(isset($_POST['login'])){
  $u_email = $_POST['email'];
  $u_pass = $_POST['password'];
  $sql = "SELECT * from users where email ='".$u_email."' AND password ='".$u_pass."' ";
  $result = mysqli_query($conn,$sql);
  $row = mysqli_fetch_array($result);
  if($row['usertype']=="user"){
    $_SESSION['user_email']=$u_email;
    $_SESSION['usertype']="user";
    header("location:../index.php"); }
  else if($row['usertype']=="admin"){
    $_SESSION['user_email']=$u_email;
    $_SESSION['usertype']="admin";
    header("location:../admin/adminpage.php"); }
  else{
    $_SESSION['message']="Username or Password is wrong";  }}
?>
```

Check out page :

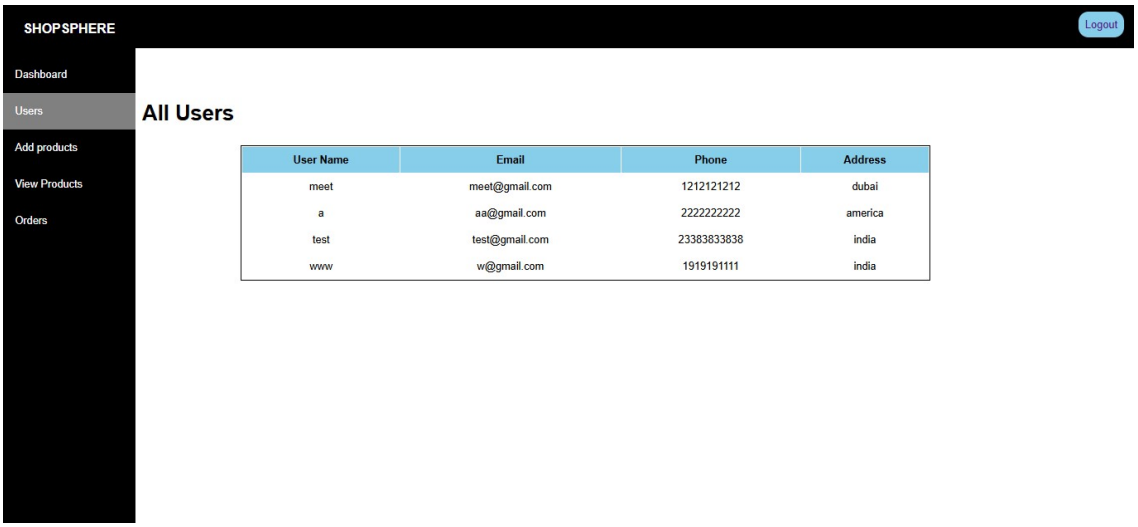

```
<?php
error_reporting(0);
$conn=mysqli_connect("localhost","root","","php_ecom");
session_start();
$my_email = $_SESSION['user_email'];
$user_email = $_GET['email'];
if($my_email){
$sql = "SELECT * from Orders where email='$my_email'";
$result = mysqli_query($conn,$sql);
}
else if($user_email){
$sql = "SELECT * from Orders where email='$user_email'";
$result = mysqli_query($conn,$sql);
}
else{
   header("location:home/login.php");
}
?>
```

Admin side :

Dashboard :

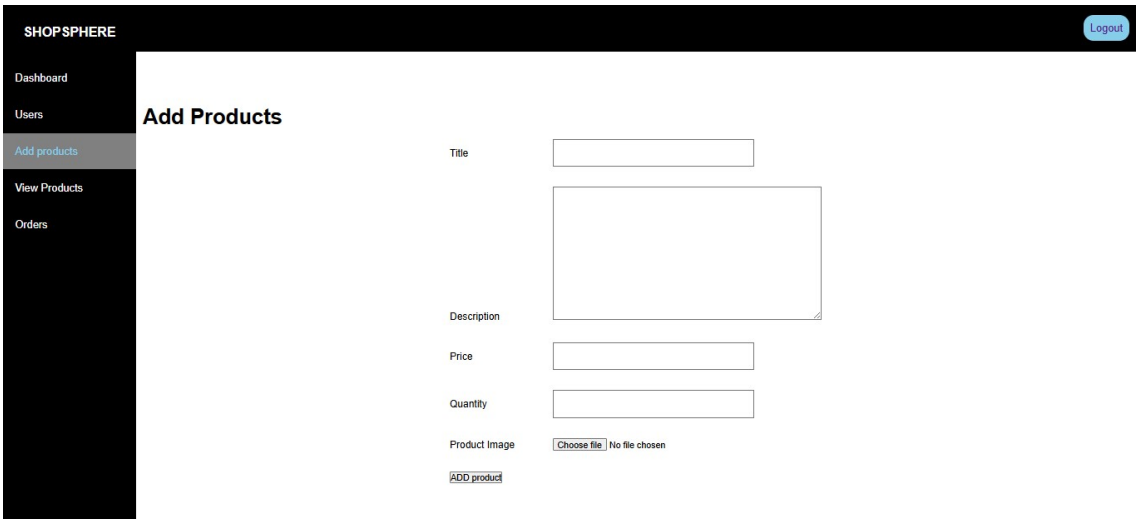

```
<?php
session_start();
$conn = mysqli_connect("localhost","root","","php_ecom");
if(!isset($_SESSION['user_email'])){header("location:../home/login.php");}
else if($_SESSION['usertype'] == "user"){header("location:../home/login.php");}
$user_sql = "Select * from users";
$u_result = mysqli_query($conn,$user_sql);
$total_user = mysqli_num_rows($u_result);/*total products*/
$product_sql = "Select * from products";
$p_result = mysqli_query($conn,$product_sql);
$total_product = mysqli_num_rows($p_result);/*total orders*/
$order_sql = "Select * from orders";
$o_result = mysqli_query($conn,$order_sql);
$total_order = mysqli_num_rows($o_result);
/*total delivered*/$deliver_sql = "Select * from orders where status = 'Delivered'";
$d_result = mysqli_query($conn,$deliver_sql);
$total_delivered = mysqli_num_rows($d_result);
?>
```

User page:



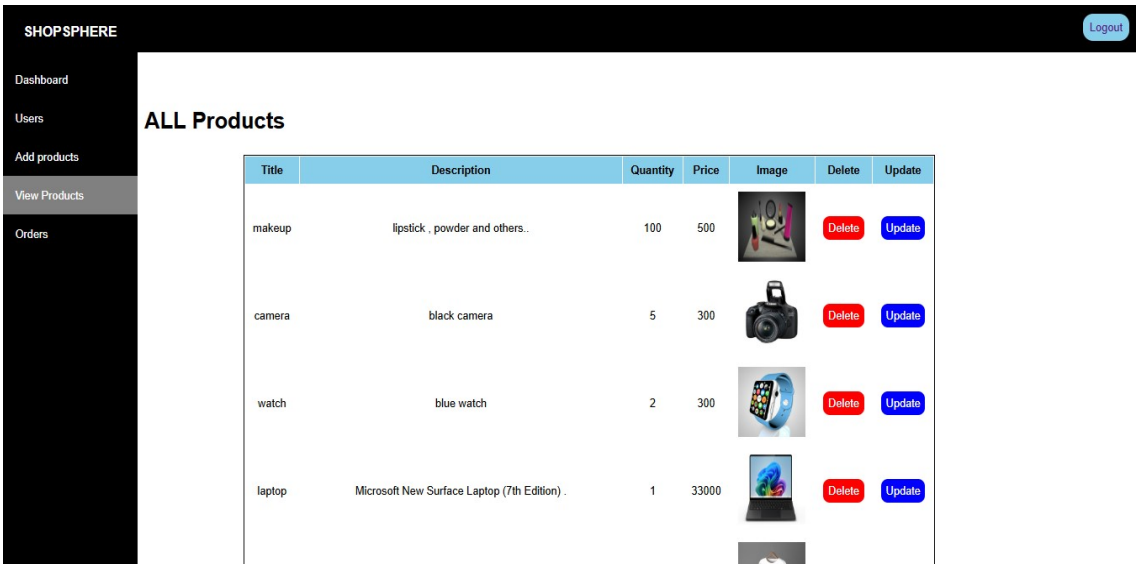

```
<?php
session_start();
$conn = mysqli_connect("localhost","root","","php_ecom");
if(!isset($_SESSION['user_email'])){
header("location:../home/login.php");}
else if($_SESSION['usertype'] == "user")
{header("location:../home/login.php");}
$is_user="user";
$sql="SELECT * from users where usertype='$is_user'";
$result=mysqli_query($conn,$sql);
?>
<?php
        while($row=mysqli_fetch_assoc($result)) {
        ?>
         <tr><td><?php echo $row['name'] ?></td>
           <td><?php echo $row['email'] ?></td>
           <td><?php echo $row['phone'] ?></td>
           <td><?php echo $row['address'] ?></td></tr>
        <?php  } ?>
```

Add product :




```
<?php
session_start();
if(!isset($_SESSION['user_email'])){header("location:../home/login.php");}
else if($_SESSION['usertype'] == "user"){header("location:../home/login.php");}
$conn=mysqli_connect("localhost","root","","php_ecom");
if(isset($_POST['add_product'])){
   $title = $_POST['title'];
   $des = $_POST['description'];
   $price = $_POST['price'];
   $qty = $_POST['qty'];
   $image_name = $_FILES['my_image']['name'];
   $tmp = explode(".",$image_name);
   $newfilename = round(microtime(true)).'.'.end($tmp);
   $uploadpath = "../product_image/".$newfilename;
move_uploaded_file($_FILES['my_image']['tmp_name'],$uploadpath);
$sql = "INSERT into products(title,description,price,quantity,image) Values
('$title','$des','$price','$qty','$newfilename')";
   $data = mysqli_query($conn,$sql);
   if($data){ header('location:add_product.php'); }}?>
```

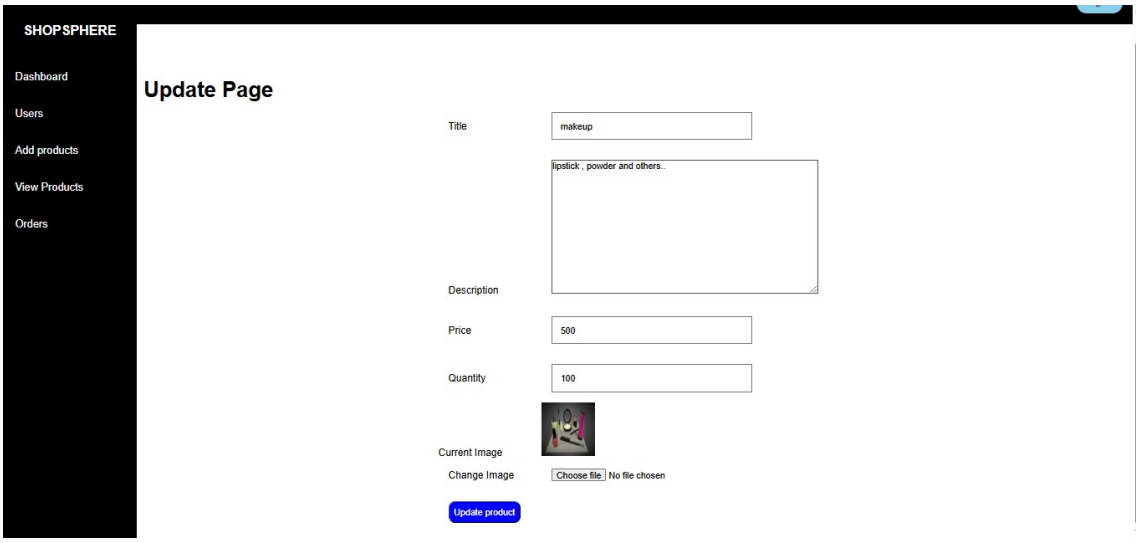
view product:




```
<?php
session_start();
if(!isset($_SESSION['user_email'])){ header("location:../home/login.php");
}
else if ($_SESSION['usertype'] =="user"){ header("location:../home/login.php");
}
$conn = mysqli_connect("localhost","root","","php_ecom");
if(isset($_GET['id'])){
  $p_id = $_GET['id'];
  $del_sql = "DELETE from products where id='$p_id'";
  $data = mysqli_query($conn,$del_sql);
  if($data){
    header("location:display_product.php");
  }
}
$sql = "SELECT * from products";
$result = mysqli_query($conn,$sql);
?>
```

```
<table><tr>
            <th>Title</th>
            <th>Description</th>
            <th>Quantity</th>
            <th>Price</th>
            <th>Image</th>
            <th>Delete</th>
            <th>Update</th>
          </tr>
          <?php
          while($row=mysqli_fetch_assoc($result))
          {
          ?>
          <tr>
            <td><?php echo $row['title'] ?></td>
            <td><?php echo $row['description'] ?></td>
            <td><?php echo $row['quantity'] ?></td>
            <td><?php echo $row['price'] ?></td>
            <td>
             <img height="100" width="100" src="../product_image/<?php echo $row['image'] ?>">
            </td><td>
            <a onclick="return confirm('Are you sure to delete this');" class="del_btn"
href="display_product.php?id=<?php echo $row['id'] ?>">Delete</a>
            </td> <td>
              <a class="upd_btn" href="update_product.php?id=<?php echo $row['id']
?>">Update</a>
            </td> </tr>
          <?php
          }
          ?>
        </table>
```

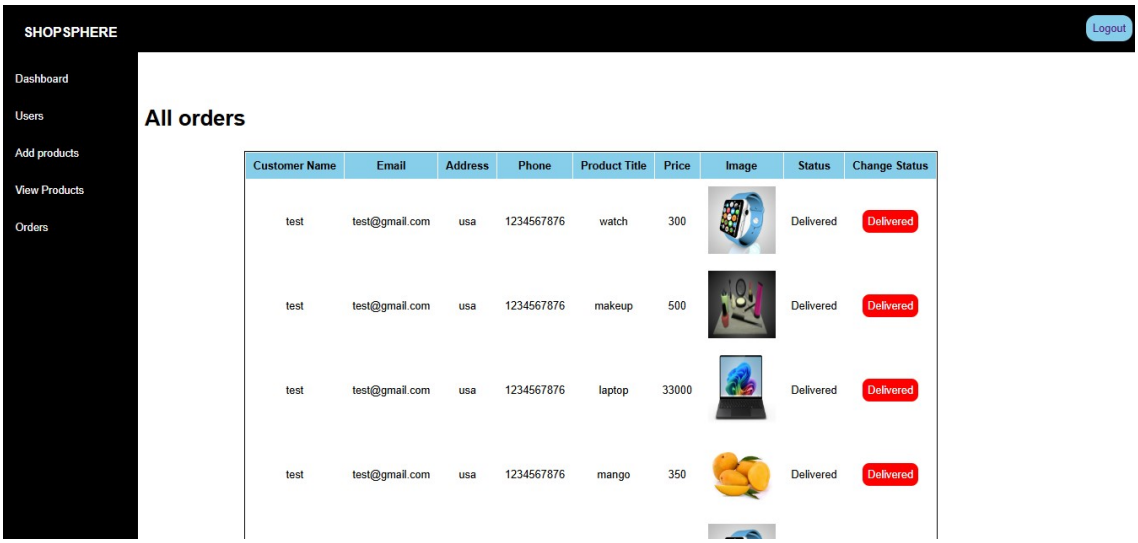
Update page :




```
<?php
session_start();
if(!isset($_SESSION['user_email'])){
   header("location:../home/login.php");
}
else if($_SESSION['usertype'] == "user")
{
   header("location:../home/login.php");
}
$conn = mysqli_connect("localhost","root","","php_ecom");
$p_id = $_GET['id'];
$sql = "SELECT * from products where id='$p_id'";
$result = mysqli_query($conn,$sql);
$row = mysqli_fetch_assoc($result);
if(isset($_POST['update_product'])){
  $p_title = $_POST['title'];
  $p_des = $_POST['description'];
```

```
  $p_price = $_POST['price'];
  $p_qty = $_POST['qty'];
  $image_name = $_FILES['my_image']['name'];
   if($image_name)
   {$tmp = explode(".",$image_name);
    $newfilename = round(microtime(true)).'.'.end($tmp);
    $uploadpath = "../product_image/".$newfilename;
   move_uploaded_file($_FILES['my_image']['tmp_name'],$uploadpath);
   $update_sql = "Update products set title='$p_title', description='$p_des',
price='$p_price', quantity='$p_qty',image='$newfilename' where id='$p_id' ";
   $data = mysqli_query($conn,$update_sql);
   if($data){
    header("location:display_product.php");
   } }
 else{
   $update_sql = "Update products set title='$p_title', description='$p_des',
price='$p_price', quantity='$p_qty' where id='$p_id' ";
   $data = mysqli_query($conn,$update_sql);
   if($data){
    header("location:display_product.php");
   }
 }
}
?>
```

Order page :




```
<?php
session_start();
$conn = mysqli_connect("localhost","root","","php_ecom");
if(!isset($_SESSION['user_email'])){
 header("location:../home/login.php");}
else if($_SESSION['usertype'] == "user")
{ header("location:../home/login.php");}
$sql = "SELECT * from orders";
$result = mysqli_query($conn,$sql);
?>
<?php
$conn = mysqli_connect("localhost","root","","php_ecom");
$order_id = $_GET['id'];
$delivered = "Delivered";
$sql = "UPDATE orders set status='$delivered' where id='$order_id' ";
$result = mysqli_query($conn,$sql);
if($result)
{  header("location:all_orders.php");}
?>
```

SPECIAL UTILITIES

Following special utilities is provide by Shopsphere Ecommerce website.

- Home Page/Landing Page: Welcomes users to the website, shows the brand "ShopSphere," and provides navigation links (Home, Products, Contact, Register, Login).
- Product Listing: Displays products (makeup, camera, watch, laptop) in cards with images, names, descriptions, prices, and an "Add to Cart" button, along with a search bar.
- Shop By Category: Allows users to browse and filter products by popular collections like Men's Clothing, Women's Wear, Accessories, and Footwear.
- User Registration: Provides a form for new users to create an account by entering their name, email, phone, address, password, and gender.
- User Login: Provides a form for existing users to securely access their accounts by entering their email and password.
- Admin Dashboard: Gives an overview of key e-commerce metrics including the Total Users, Total Products, Total Orders, and Total Delivered items.
- View All Users: Displays a complete list of all registered users with their User Name, Email, Phone number, and Address.
- Add Products: Provides a form for the administrator to upload new products by entering the Title, Description, Price, Quantity, and Product Image.
- View Products/Product Management: Lists all products and allows the admin to Delete or Update (edit) product details like Title, Description, Quantity, and Price.
- All Orders Management: Displays a list of customer orders with details (Customer Name, Product Title, Price, Status) and a function to Change Status (e.g., to 'Delivered').

TESTING

Software Development Life Cycle (SDLC) includes a series of production activities one of this is testing.

Testing is a process of executing a program with the intent of finding an error.

Testing is the most important element to be considered for providing quality software and it represents the ultimate review of specification, design and coding.

The success or failure of the software as a system mainly depends on testing. Software Developer spends 40% to 50% of their total development time on testing.

There are several SDLC techniques and development model. I have focused on Prototype Model. I have followed the prototyping model to develop this Software.

The development of software system involves a series of production activities where opportunities for injection of human fallibility are enormous. Error may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as later design and development states. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Testing is program consists of providing the program with a set of test inputs and observing if the programs behave as expected. Under which a failure occurs are noted for debugging and correction. The following are some commonly used terms associated with testing.

A failure is manifestation of an error. But, the mere presence of an error may not necessarily lead to a failure.

A fault is an incorrect intermediate state that may have been entered during program execution. A fault may or may not lead to a failure.

A test case is the triplet [I.S.O.], where I is the data input to the system, S is the state of the system at which the data is input, and O is the expected output of the system.

A test suite is the set of all test cases with which a given software product is to be tested.

Many types of testing techniques are describes as follows.

Unit Testing:

Unit testing is under taken when a module has been coded and successfully reviewed in this section we first discuss the environment needed to perform unit testing.

Here in this project we test each and every module and forms of software application individually when it is completely coded.

There are some methods for unit testing are as follows.

Black-Box Testing:

- **Equivalence Class Partitioning**
- **Boundary Value Analysis**

White-Box Testing:

- **Statement coverage**
- **Branch Coverage**
- **Condition Coverage**
- **Path Coverage**
- **Linearly independent Path**
- **Data Flow - Based Testing**
- **Mutation testing**

Integration Testing:

The primary objectives of the integration testing is to test the module interface in order to ensure that there are no error in parameter passing when one module invokes another module.

During integration testing different module of system as per integration plane the integration plan specify the steps and the order in which module are combine to realize the full system.

After each integration test the practical integrated system is tested.

Following are the integration testing Methods & Approaches

- **Big bang approach**
- **Top down approach**
- **Bottom up approach**
- **Mixed approach**

System Testing:

In the system testing the whole application is tested and the error and failure possibility is carried out in it.

Following are the method & approach of system testing.

- **Alpha testing**
- **Beta testing**
- **Acceptance testing**
- **Performance testing**
- **Error seeding**

Testing is a process of executing a program with the intent of finding an error.

A good test case is one that as a high probability of finding an as yet in discovered error.

A successful test is one that uncovers a yet undiscovered error.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. the increasing visibility of software as a system element and the attendant "Cost" associated with a software failure are motivating forces for well – planned, thorough testing. It is not unusual for a software development organization to expend between 30 to 40 percent of total project effort on testing. In the extreme, testing of human-rated software can cost three to five times as much as all other software engineering activities combined.

IMPLEMENTATION

Implementation refers to the entire effort associated with a new system. The implementation of a web application involves longer term issues after the system has been designed and installed. Implementation is a part of the design of a web application, and is an organizational change process. It is a part of the process that begins with the very first idea for a web application has been successfully integrated with the operations of the organization. We expect most of the implementation to be concerned with behavioral phenomena since people are expected to change their information processing activities.

The implementation is processed from review and reports from developer cover the following areas:

- **Good working conditions.**
- **Useful for gathering information.**
- **Update website easily.**
- **Attractive layouts.**
- **Working for as per requirements.**

BIBLIOGRAPHY

Websites:

- https://www.youtube.com/
- https://www.microsoft.com/
- https://www.google.com/