KHADIMART

A Project Report

Submitted for the partial fulfillment for the award of the degree of

B. Tech CSE 3rd Year VI Semester

Submitted by

LAVANYA BHARGAVA: 2116209
MIMANSHA SHARMA: 2116222
OJASVI SHANKHDHAAR: 2116241

Under the supervision of

Dr. AJAY KUMAR YADAV

Department of Computer Science
Banasthali Vidyapith

Session: 2023-24

Certificate

Certified that **Lavanya Bhargava, Mimansha Sharma and Ojasvi Shankhdhaar** has carried out the project work titled **KHADIMART** from **01 August 2024** to **31 March 2024** for the award of the **B. Tech 3rd Year CSE** from **BANASTHALI VIDYAPITH** under my supervision. The thesis embodies result of original work and studies carried out by Student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Dr. AJAY KUMAR YADAV

Designation: Assistant Professor

Place: Banasthali Vidyapith

Date:07 April,2024

Abstract:

This report covers the journey of creating Khadimart, an online store specifically designed for Khadi products. We discuss how we planned and built the website, including what tools and technologies we used, and how we made sure it would work well. We also look at the different features of the website, like how users can create accounts, browse products easily, and make purchases smoothly. By explaining the technical, economic, and practical sides of Khadimart, we show how it can provide a great shopping experience for users while also helping Khadi artisans and businesses grow. This report is a roadmap to understanding how Khadimart came to be and how it can make a positive impact on the Khadi industry and culture.

Acknowledgement:

We are profoundly grateful to our mentor, Dr. Ajay Kumar Yadav, whose unwavering support, invaluable guidance, and insightful feedback have been instrumental in shaping this project. We are deeply appreciative of the opportunity to learn from you and for the invaluable mentorship you have provided throughout this journey.

We would also like to extend our gratitude to Banasthali Vidyapith for providing the resources and environment for project.

Additionally, we express our heartfelt thanks to our project coordinators Dr. Deepak Kumar and Dr Neelam Sharma for their indispensable support and coordination throughout the duration of this project. Their organization, guidance, and assistance have been pivotal in ensuring the smooth progress and successful completion of our project. We greatly appreciate their dedication and commitment to facilitating our work and contributing to its overall success.

TABLE OF CONTENTS:

Introduction:

Khadimart is an ecommerce website where you can effortlessly explore, search for, and purchase a wide array of Khadi items. We've designed Khadimart with simplicity in mind, ensuring that everyone can navigate it with ease. With features like easy registration and a straightforward payment-on-delivery system, we prioritize convenience for both buyers and sellers alike. Additionally, Khadimart serves as a platform for Khadi artisans and entrepreneurs to reach a broader audience, thereby supporting and promoting the Khadi industry. Our mission is to make Khadi more accessible to people everywhere, fostering its sustainability and cultural significance. When you shop on Khadimart, you not only acquire quality products but also contribute to the preservation and growth of Khadi.

Requirement analysis (SRS):

1. Requirement Specification:

- User Accounts: The Khadimart website will feature robust user registration and login functionalities, allowing users to create accounts, manage their profiles, and conveniently track their orders.
- Product Listings: The website will showcase an extensive array of products meticulously categorized for easy navigation and enhanced user experience.
- Shopping Cart: Users will have the ability to add items to their shopping carts, modify quantities as needed, and seamlessly proceed to checkout for a hassle-free shopping experience.
- Search Functionality: Our website will boast a powerful search feature, empowering users to effortlessly find products using keywords, categories, and various filters, thus enhancing their shopping experience.
- Order Management: Admins will have privileged access to an intuitive order management system, enabling them to efficiently track orders, update order statuses, and manage inventory to ensure seamless operations.

- Responsive Design: Khadimart is committed to delivering an optimal user experience across all devices and screen sizes by implementing a responsive design, allowing users to access the website conveniently from desktops.
- Admin Panel: The website will feature a comprehensive admin panel equipped with intuitive tools for managing products, categories, users, orders, and other site content, enabling administrators to efficiently oversee and maintain the platform.

2. H/w and S/w Requirements:

Hardware Requirements:

- Web Server: A basic computer with moderate specifications can serve as a web server for hosting the Khadimart website. A dual-core processor, 4GB of RAM, and sufficient storage capacity (at least 250GB HDD) should be adequate for handling the anticipated traffic load.
- Database Server: Similar to the web server, a basic computer with specifications comparable to the web server can be used as a database server. It should have sufficient processing power and storage to efficiently manage the database. Again, a dual-core processor, 4GB of RAM, and ample storage space (250GB HDD) should suffice.
- Storage Solutions: Standard hard disk drives (HDD) are sufficient for storage needs at this level. However, consideration should be given to backup solutions to ensure data integrity and recovery in case of system failures.
- Network Infrastructure: A stable internet connection with sufficient bandwidth is essential for hosting the website and facilitating smooth user interactions. A broadband connection with at least 10 Mbps upload and download speeds should be adequate for initial deployment.

Software Requirements:

- Web Development Frameworks: XAMPP, a comprehensive software stack that includes Apache, MySQL, PHP, and Perl, can be utilized for local development and testing purposes. These frameworks provide an integrated environment for developing and testing PHP-based web applications.
- Database Management System (DBMS): MySQL, included in XAMPP, can serve as the primary database management system for storing and retrieving data related to products, users, orders, etc. It's widely used, easy to learn, and well-supported within the PHP development ecosystem.
- Front-end Technologies: HTML, CSS, and JavaScript are the primary front-end technologies being used for developing the Khadimart website. These languages are fundamental for creating the user interface, styling elements, and adding interactive features to the website.

3. Feasibility Study:

- Technical Feasibility: The Khadimart project is technically feasible due to the availability of widely-used web development frameworks such as XAMPP, which provides an integrated environment for PHP-based web development. Additionally, the availability of payment gateway APIs allows for secure online transactions. The use of HTML, CSS, JavaScript, and PHP technologies provides a solid foundation for implementing the required functionalities.
- Economic Feasibility: The project's economic feasibility is favourable, considering various factors such as development costs, maintenance costs. By utilizing open-source technologies and frameworks like XAMPP, development costs can be kept relatively low. Additionally, the decision to offer cash on delivery as a payment option mitigates risks associated with online payment processing, potentially increasing customer confidence and boosting revenue.
- Operational Feasibility: The project aligns with Khadimart's goal of providing a seamless and convenient shopping experience for customers

while maintaining operational efficiency. The decision to integrate cash on delivery as a payment option ensures that the website's operational processes are compatible with existing practices. By offering this payment method, Khadimart can cater to a wider audience, including those who prefer traditional payment methods or may have concerns about online payment security.

4. Product Functions:

- User Registration and Login: Khadimart provides users with the ability to register for an account on the website, enabling them to access exclusive features such as adding items to the cart and Wishlist.
  Upon registration, users can securely log in to their accounts using their credentials, ensuring a personalized and convenient shopping experience.
  The user registration and login functionality of Khadimart allows customers to create accounts.
- Product Browsing and Search: Khadimart offers users a seamless browsing experience, allowing them to explore a diverse range of products showcased on the website.
  A robust search feature is integrated, enabling users to easily find products based on keywords, categories, or specific filters.
  The product browsing and search functionalities of Khadimart empower users to discover and explore a wide selection of artistic products with ease.
- Shopping Cart Management: Users can add desired items to their shopping carts, modify quantities as needed, and review their selections before proceeding to checkout.
  The shopping cart functionality ensures a smooth and hassle-free shopping experience, allowing users to manage their purchases conveniently.
  Khadimart's shopping cart management system enables users to curate their selections and finalize their purchases seamlessly, enhancing the overall shopping journey.

- Secure Payment Processing (Cash on Delivery): Khadimart offers the option for users to pay for their purchases using cash on delivery (COD) method.
  Users can select the COD option during checkout, and upon successful delivery of the products, they can make the payment in cash to the delivery personnel.
  As part of Khadimart's payment methods, the cash on delivery option provides users with a convenient and trustworthy way to complete their purchases, especially for those who prefer not to make online payments.
- Admin Panel for Site Management: Khadimart administrators have access to a comprehensive admin panel, providing them with tools and functionalities to manage various aspects of the website.
  Admins can oversee product listings, manage categories, monitor user activity, process orders, and perform other administrative tasks efficiently.
  The admin panel of Khadimart serves as a centralized hub for site management, empowering administrators to maintain and optimize the website's performance and functionality.

5. Use-case Diagrams:

Actors:

Admin: Admin is responsible for website management, which includes user authentication, category management, item management, and order management.

User: Users are customers who interact with the website to register, view items, make orders, make payments, and change their passwords.

Admin Operations:

1. Admin Login: Admin logs in to access the admin dashboard.

2. Add Category: Admin can add new product categories to the website.

3. Manage Items: Admin can manage product items, including adding, editing, and removing items.

4. Manage Orders: Admin can view and manage customer orders, including order details and status updates.

User Operations:

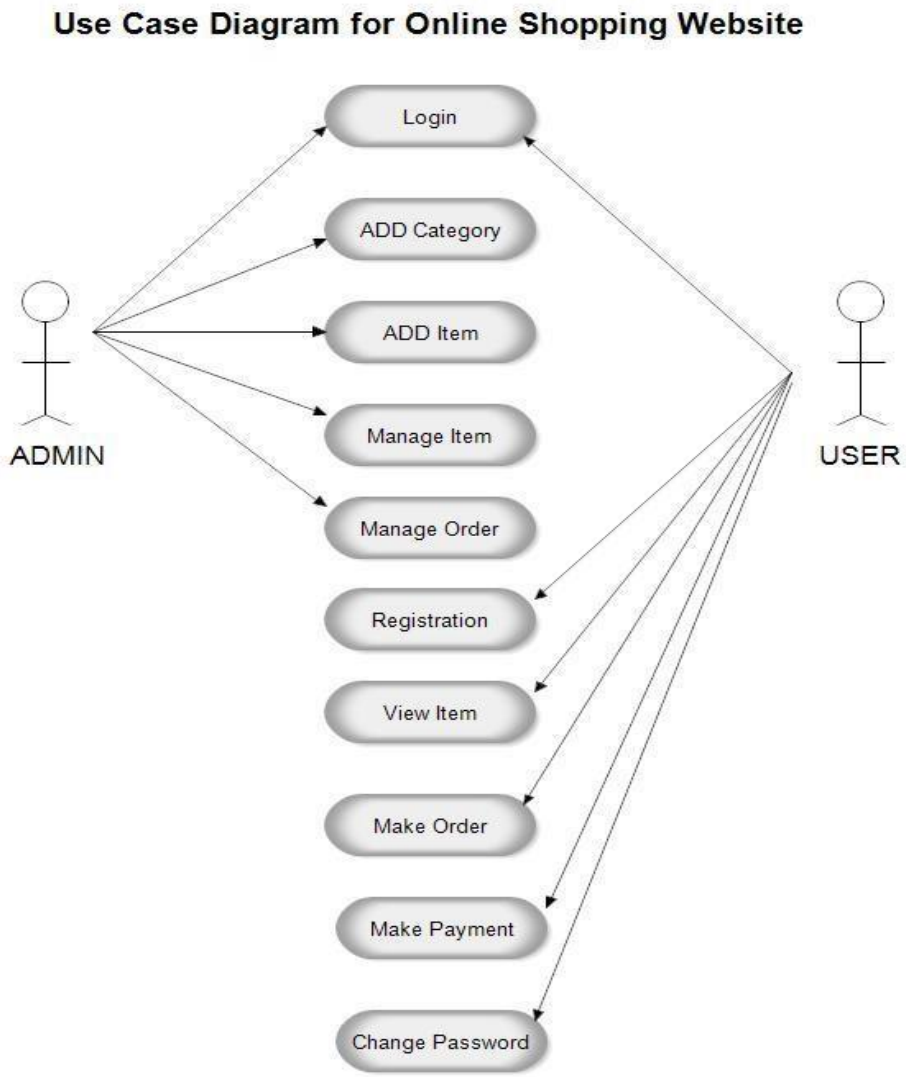
1. User Registration: Users can create accounts on the Khadimart website.

2. View Items: Users can browse and view product items.

3. Make an Order: Users can select items, add them to the shopping cart, and place orders.

4. Make a Payment: Users can proceed to make payments for their orders.

5. Change Password: Users can change their account passwords.



System Design (SDS):

1. High-level Design

Khadimart is designed as a modern and efficient online shopping platform, integrating several key components to provide a seamless and secure shopping experience for users.

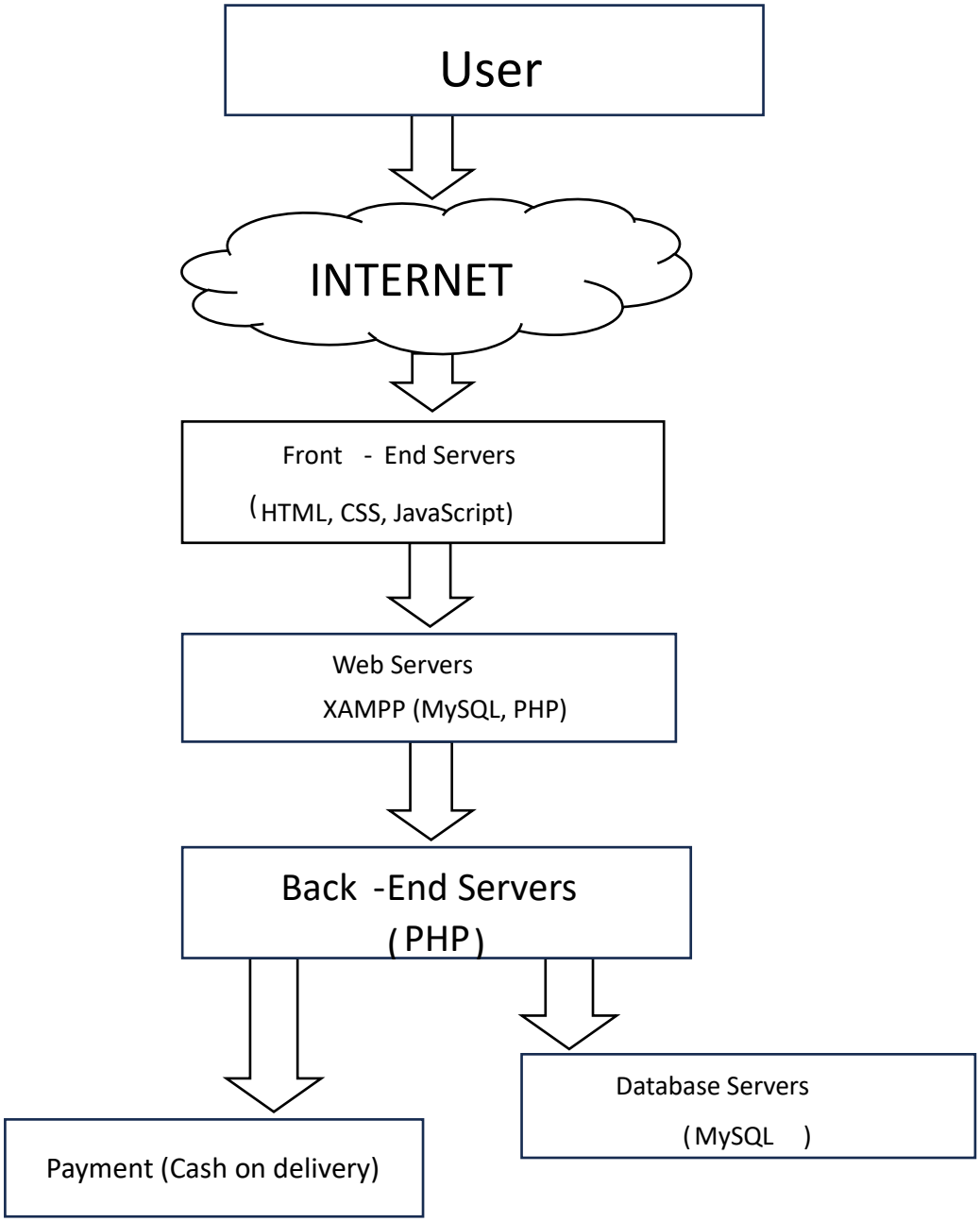
- Front-End Components: HTML/CSS/JavaScript: These technologies are utilized for creating the website's structure, styling, and interactivity, ensuring a visually appealing and user-friendly interface.
- Web Servers: XAMPP (MySQL/PHP): XAMPP serves as the web server environment, utilizing Apache for HTTP request handling and PHP for server-side scripting. It acts as the gateway for user interactions, facilitating dynamic content delivery and processing user inputs.
- Back-End Components: The back-end logic is implemented in PHP, handling user authentication, order processing, and interaction with the database.
- MySQL: MySQL database is used to store and manage essential data such as product details, user accounts, and order history. PHP interacts with MySQL through MySQL or PDO for data retrieval and updates.
- Payment Processing: Cash on Delivery Option: In addition to online payment options, Khadimart offers a cash on delivery (COD) option for payment processing. This allows users to make payments in cash upon delivery of their orders, providing a convenient and trusted payment method for those who prefer not to transact online.

High-level System Architecture:

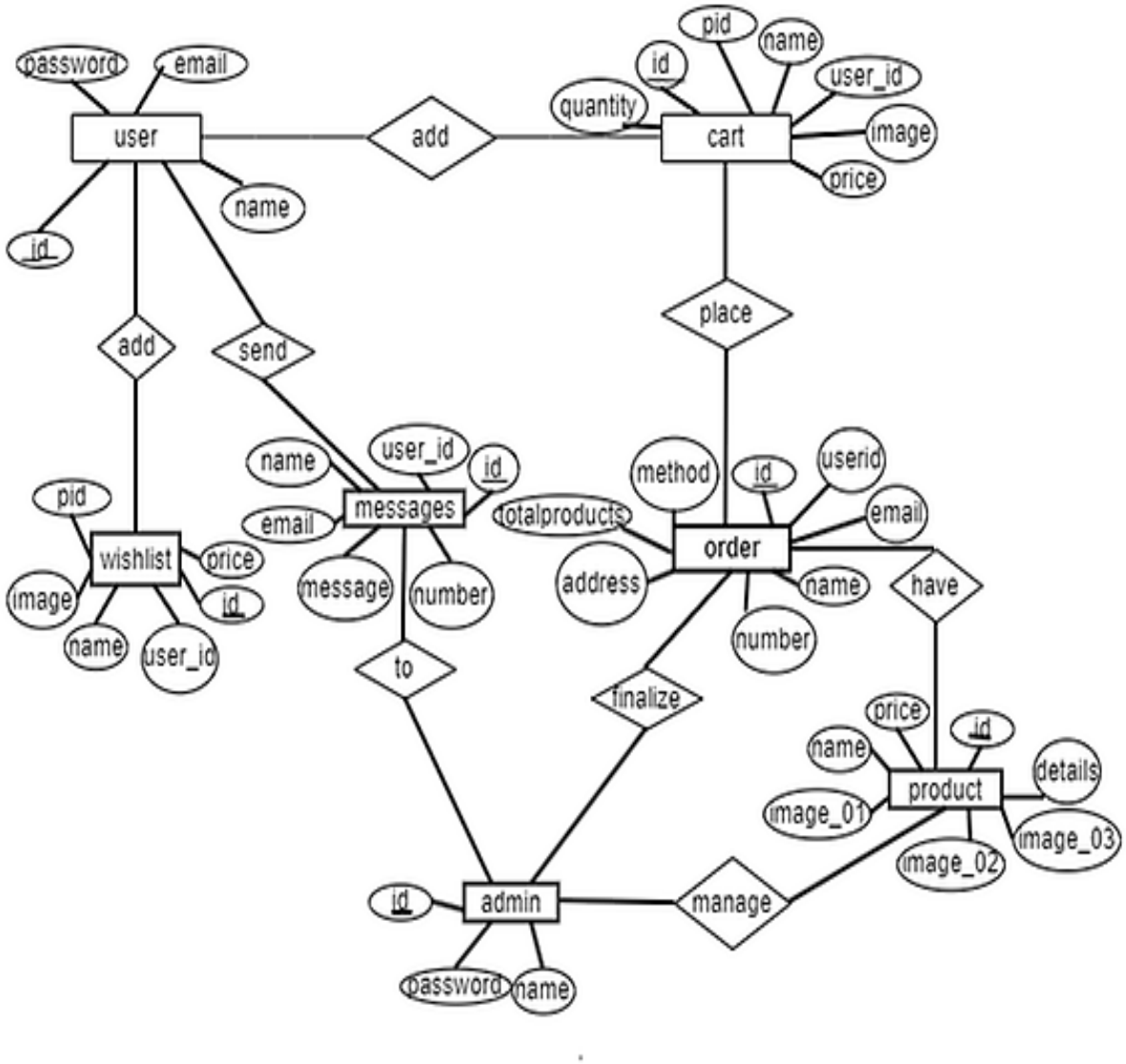
The high-level system architecture diagram showcases the core components of Khadimart, illustrating how they interact to deliver a seamless shopping experience to users.

It highlights the flow of data and control between the front-end, back-end, database, and payment demonstrating the integrated nature of the system.

The inclusion of the cash on delivery option alongside other payment methods reflects Khadimart's commitment to catering to diverse user preferences and ensuring a convenient and secure shopping experience for all customers.



2. ER Diagram:



Users: This table stores information about users of the system.

id (Primary Key): A unique identifier for each user.

name: The name of the user.

email: The email address of the user.

password: The password of the user for authentication.

Admins: Similar to the Users table but specifically for administrators who manage the system.

id (Primary Key): A unique identifier for each admin.

name: The name of the admin.

password: The password of the admin for authentication.

Products: This table contains details about the products available in the system.

id (Primary Key): A unique identifier for each product.

name: The name of the product.

details: Additional details or description of the product.

price: The price of the product.

image_01, image_02, image_03: URLs or references to images representing the product.

Cart: Represents the shopping cart of users, containing items added by users for purchase.

id (Primary Key): A unique identifier for each item in the cart.

user_id (Foreign Key referencing Users): Identifies the user who added the item to the cart.

pid (Foreign Key referencing Products): Identifies the product added to the cart.

name: The name of the product in the cart.

price: The price of the product in the cart.

quantity: The quantity of the product added to the cart.

image: URL or reference to the image representing the product.

Wishlist: Stores products that users have added to their Wishlist for future consideration or purchase.

id (Primary Key): A unique identifier for each item in the Wishlist.

user_id (Foreign Key referencing Users): Identifies the user who added the item to the Wishlist.

pid (Foreign Key referencing Products): Identifies the product added to the Wishlist.

name: The name of the product in the Wishlist.

price: The price of the product in the Wishlist.

image: URL or reference to the image representing the product.

Orders: Contains information about orders placed by users.

id (Primary Key): A unique identifier for each order.

user_id (Foreign Key referencing Users): Identifies the user who placed the order.

name: The name associated with the order (e.g., recipient name).

number: A unique order number or identifier.

email: The email address associated with the order.

method: The payment method used for the order.

address: The shipping address for the order.

Total_products: The total number of products in the order.

total_price: The total price of the order.

Placed_on: The timestamp or date when the order was placed.

payment_status: The status of payment for the order.

Messages: Stores messages sent by users.

id (Primary Key): A unique identifier for each message.

user_id (Foreign Key referencing Users): Identifies the user who sent the message.

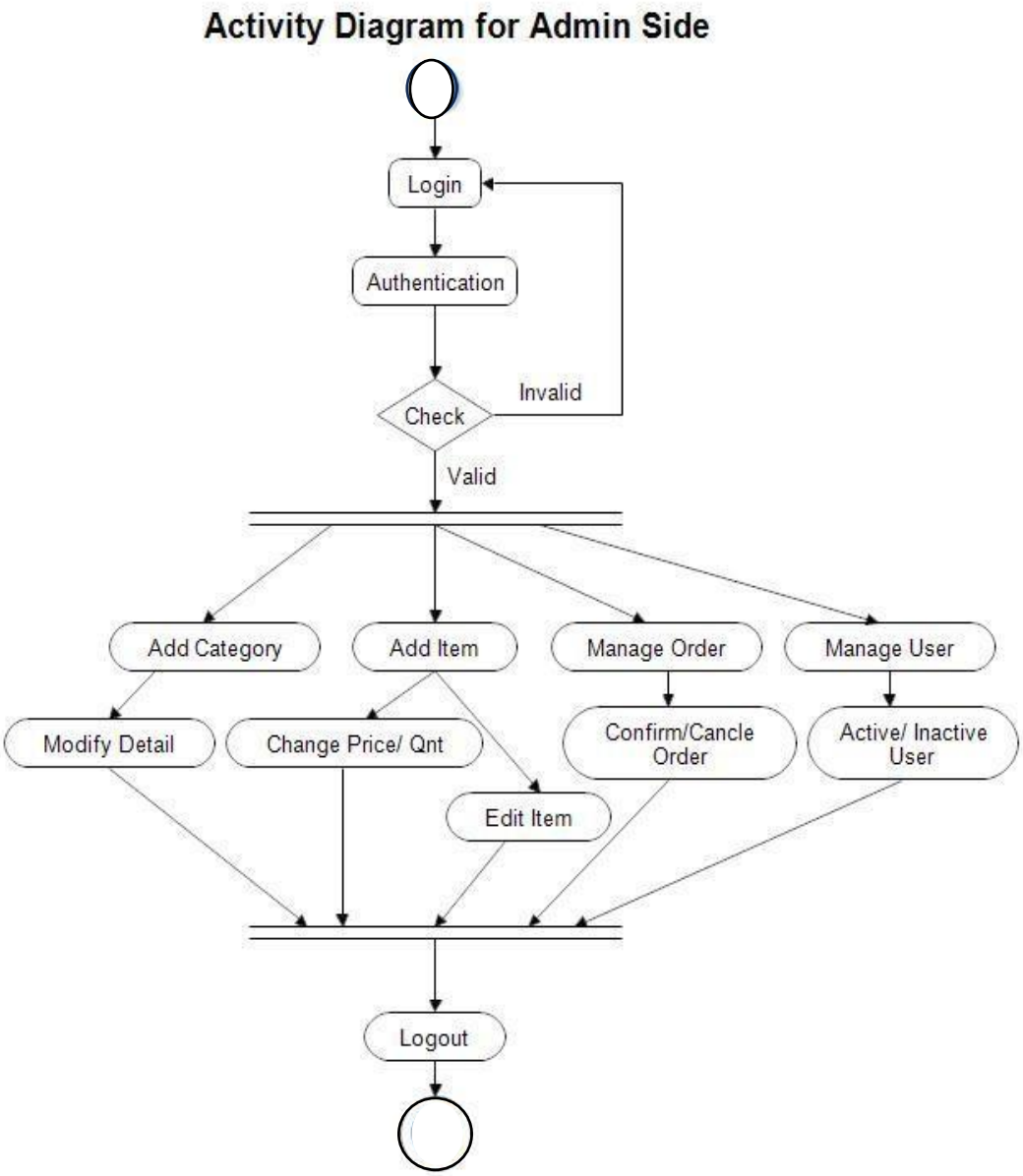
name: The name of the sender.

email: The email address of the sender.

number: The contact number of the sender.

message: The content of the message.

3. Activity Diagram:



Description:

1. Login: The activity begins with the "Login" action, where the administrator enters their credentials (username and password) and initiates the login process.

2. Authentication: After the login action, the system performs authentication to verified the administrator's identity.

If authentication is successful, the process proceeds to the "Authentication Successful" decision point. If authentication fails, the process loops back to the "Login" action.

Authentication Successful:

When the administrator is successfully authenticated, they can access various admin-side operations.

Manage Categories: The admin can choose to "Add Category" or "Modify

Category."

Add Item: The admin can add new items to the e-commerce platform.

Manage Orders: The admin can "Manage Orders" to process, update, or

Manage Users: The admin can "Manage Users," which may involve actions like account suspension, modification, or reviewing user data.

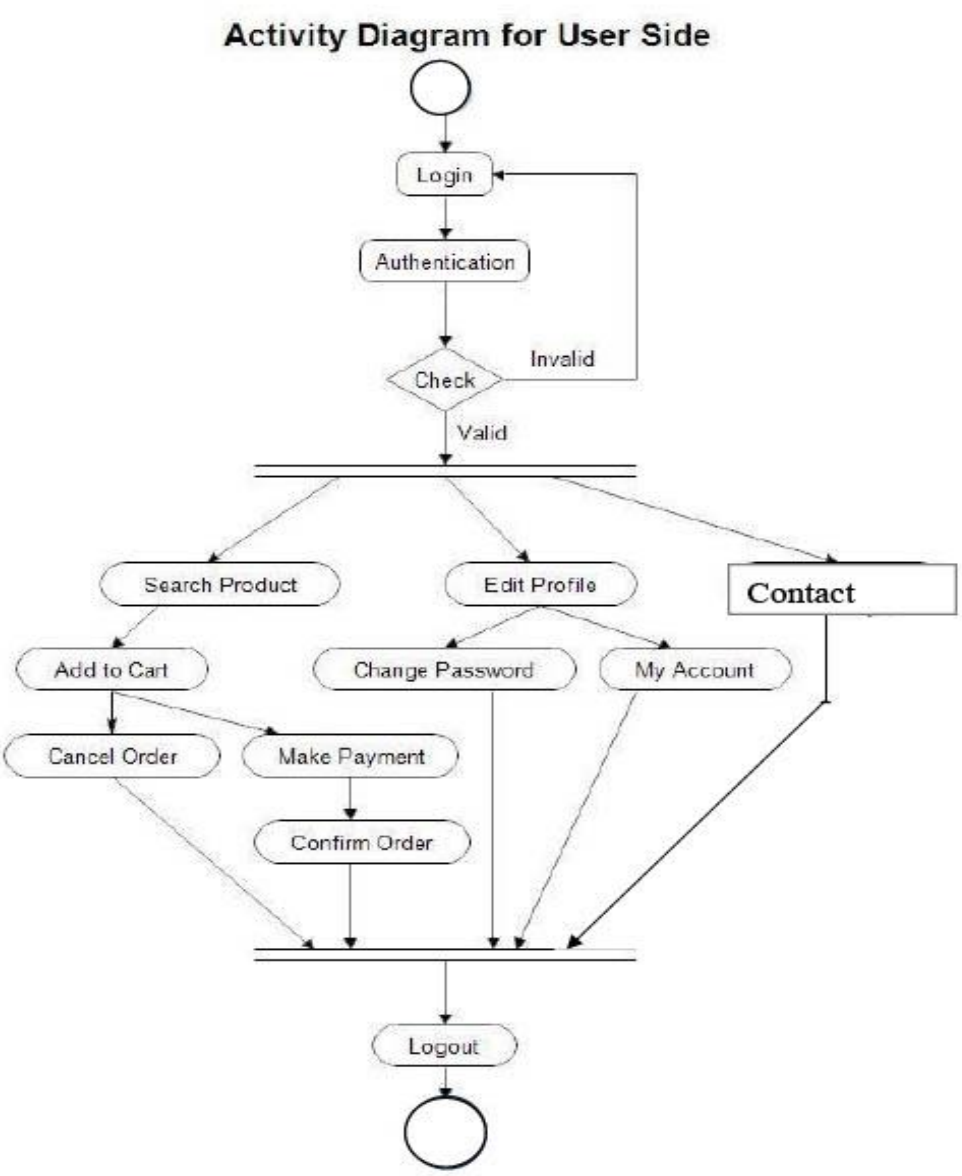
Modify Data: The admin can make modifications to the website's data, which could include updating product details, categories, or other information.

3. Logout: The admin can choose to "Logout" to end the admin session.

After logging out, the process ends.

Invalid Authentication:

If authentication fails, the process returns to the "Login" action, allowing the administrator to make another login attempt.


Activity Diagram for User Side
Login
Authentication
Check
Invalid
Valid
Search Product
Edit Profile
Contact
Add to Cart
Change Password
My Account
Cancel Order
Make Payment
Confirm Order
Logout

Description:

1. Login: The activity begins with the "Login" action, where the user enters their credentials (username and password) and initiates the login process.

2. Authentication: After the login action, the system performs authentication to verified the user's identity.

If authentication is successful, the process proceeds to the "Authentication Successful" decision point. If authentication fails, the process loops back to the "Login" action.

Authentication Successful: When the user is successfully authenticated, they can access various user-side operations.

Search Product: The user can search for products on the e-commerce platform.

After searching for a product, the process proceeds to the "Add to Cart" action.

Add to Cart: The user can add products to their shopping cart.

After adding to the cart, the process offers two options:

"Cancel Order": The user can choose to cancel the order from the cart.

"Make Payment": The user can proceed to make a payment.

Cancel Order: The user can cancel an order from the shopping cart.

After cancelling the order, the process returns to the "Add to Cart" action.

Make Payment: The user can initiate a payment for the selected products.

After making the payment, the process proceeds to the "Confirm Order" action.

Confirm Order: The user can confirm their order, which includes the final review of the purchase.

After confirming the order, the process proceeds to the "Logout" action.

Edit Profile: The user can choose to "Edit Profile," which includes options like changing their password or managing their account.

After editing the profile, the process proceeds to the "Logout" action.

Change Password: The user can change their password to enhance account security.

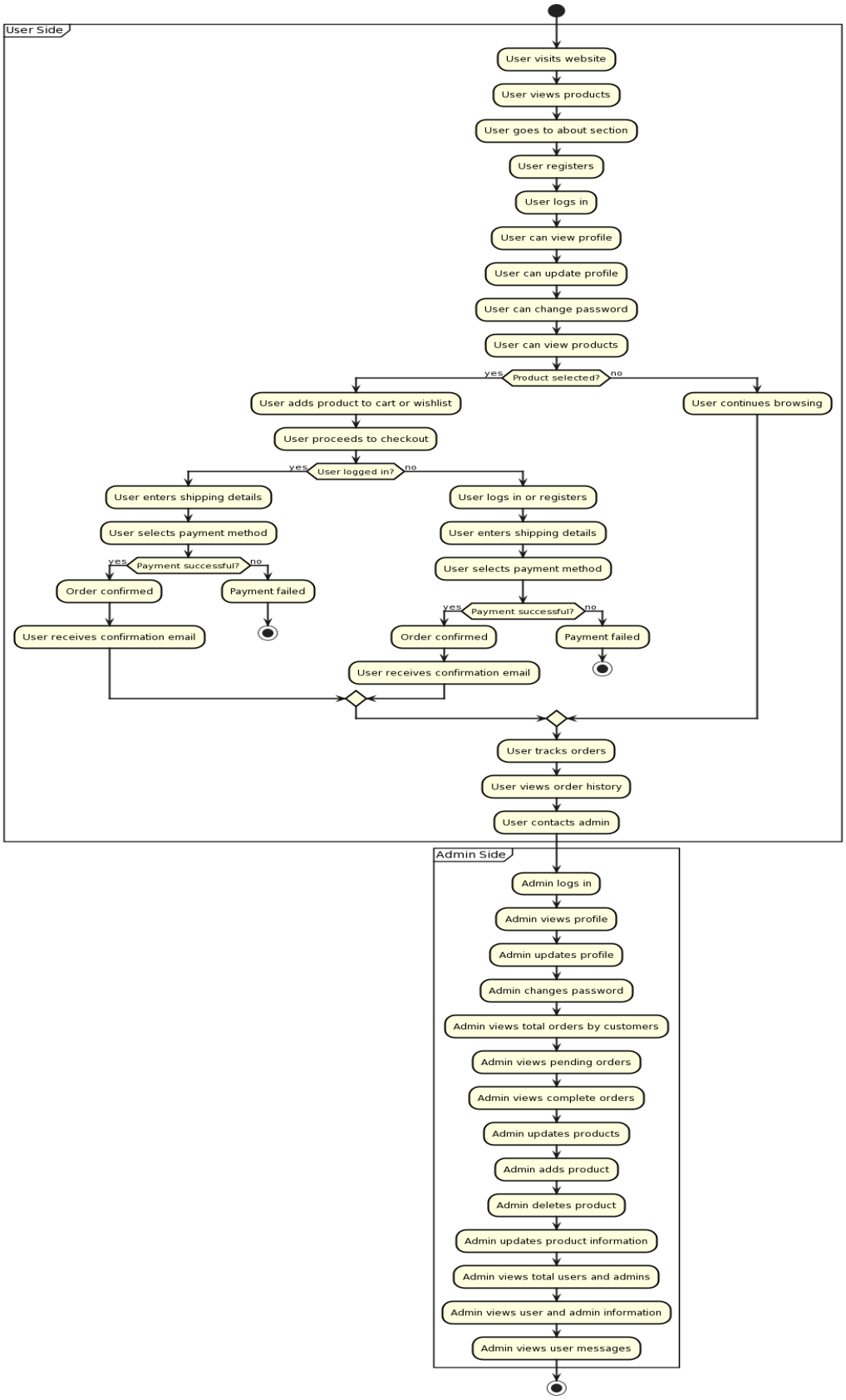
After changing the password, the process returns to the "Edit Profile" action.

3. Logout: The user can choose to "Logout" to end their session.

After logging out, the process ends.

4. Invalid Authentication: If authentication fails, the process returns to the "Login" action, allowing the user to make another login attempt.

4. Flow chart:


User Side
User visits website
User views products
User goes to about section
User registers
User logs in
User can view profile
User can update profile
User can change password
User can view products
Product selected?
yes
no
User adds product to cart or wishlist
User continues browsing
User proceeds to checkout
User logged in?
yes
no
User enters shipping details
User selects payment method
Payment successful?
yes
no
Order confirmed
Payment failed
User receives confirmation email
User logs in or registers
User enters shipping details
User selects payment method
Payment successful?
yes
no
Order confirmed
Payment failed
User receives confirmation email
User tracks orders
User views order history
User contacts admin
Admin Side
Admin logs in
Admin views profile
Admin updates profile
Admin changes password
Admin views total orders by customers
Admin views pending orders
Admin views complete orders
Admin updates products
Admin adds product
Admin deletes product
Admin updates product information
Admin views total users and admins
Admin views user and admin information
Admin views user messages

- **User Side:**

  Registration and Login: Users can register themselves by providing necessary details.

  Registered users can log in using their credentials to access the features of the website.

  Product Browsing: Products are visible to all visitors without the need for registration or login. Users can browse through various products available on the website.

  Cart and Wishlist: Users can add products to their cart for purchasing. Additionally, they can maintain a Wishlist for future reference.

  Order Management: Users can view all their previous and current orders. They can track the status of their orders (e.g., pending, completed).

  Profile Management: Users can update their usernames and change passwords as needed.

  Contact Admin: Users have the option to contact the admin for queries or support through a designated contact option.

- **Admin Side:**

  Profile Management: Admins can update their profile information and change passwords securely.

  Order Tracking: Admins can access information on total orders made by every customer. They can differentiate between pending and completed orders.

  Product Management: Admins have the authority to add, delete, or update product information. This includes features like adding new products, updating existing ones, or removing outdated listings.

  User and Admin Management: Admins can view data on the total number of users and admins registered on the platform. They can access and manage user and admin profiles, including account information.

  Message Handling: Admins can view and respond to messages sent by users through the contact option.

- **Additional Functionalities:**

  About Section: A dedicated section providing information about the website's purpose, mission, and any other relevant details.

  Security Measures: Implement security measures such as encryption, secure authentication, and data protection protocols to ensure user and transactional data safety.

  Responsive Design: Ensure the website is responsive and accessible across different devices and screen sizes for a seamless user experience.

  Search Functionality: Implement a search feature allowing users to quickly find specific products based on keywords or categories.

  COD Specifics:

  Order Confirmation for Admin: Admins receive notifications of new COD orders.

  Payment Handling: Admins manage cash received upon delivery.

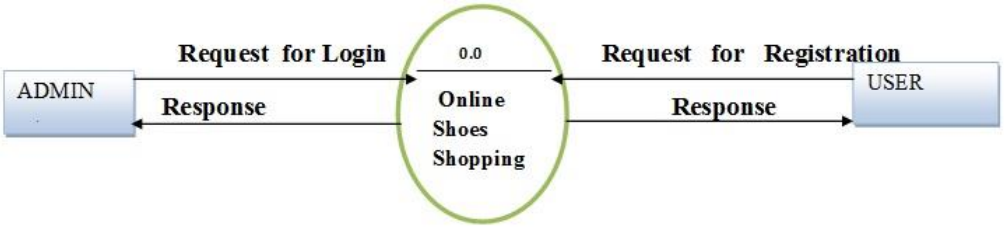
  Verification: Delivery personnel verified the COD order upon delivery.

  Order Completion: Once the COD order is delivered and payment received, admins mark the order as complete.

5. Data Flow Diagrams:

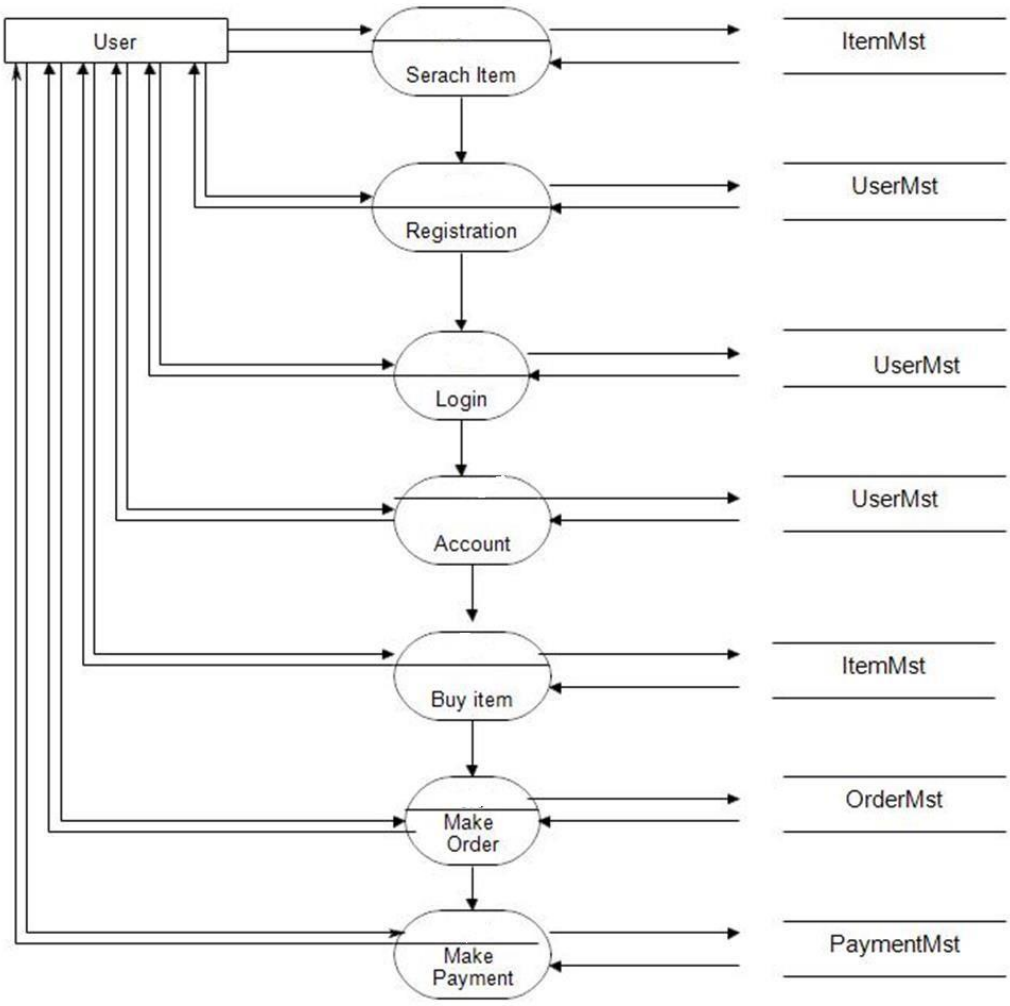
Level 0 DFD:

The context level data flow diagram (DFD) describes the whole system. The 0 level DFD describe the all-user module who operate the system. Below data flow diagram of online shopping site shows the two users can operate the system Admin and Member user.

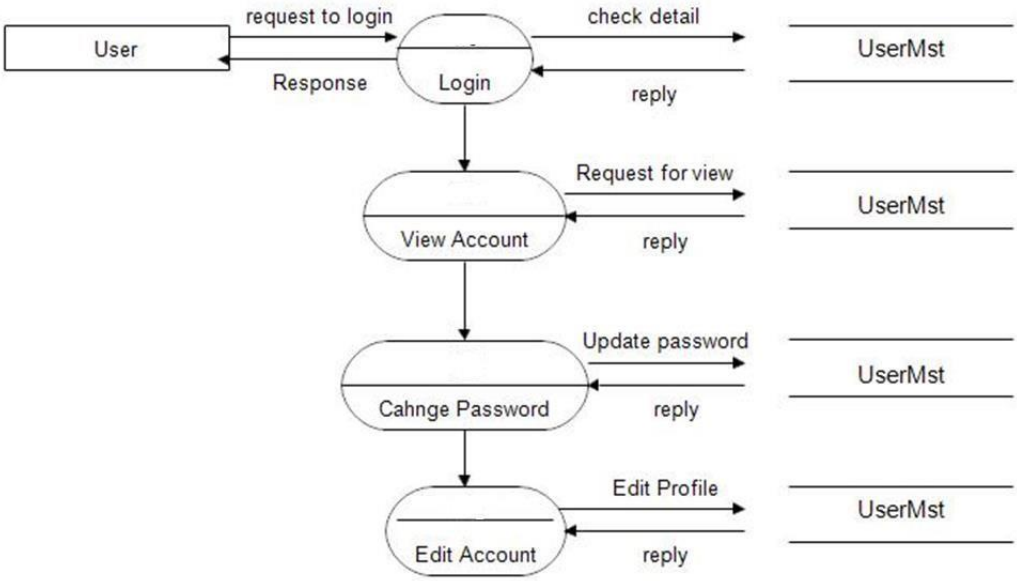


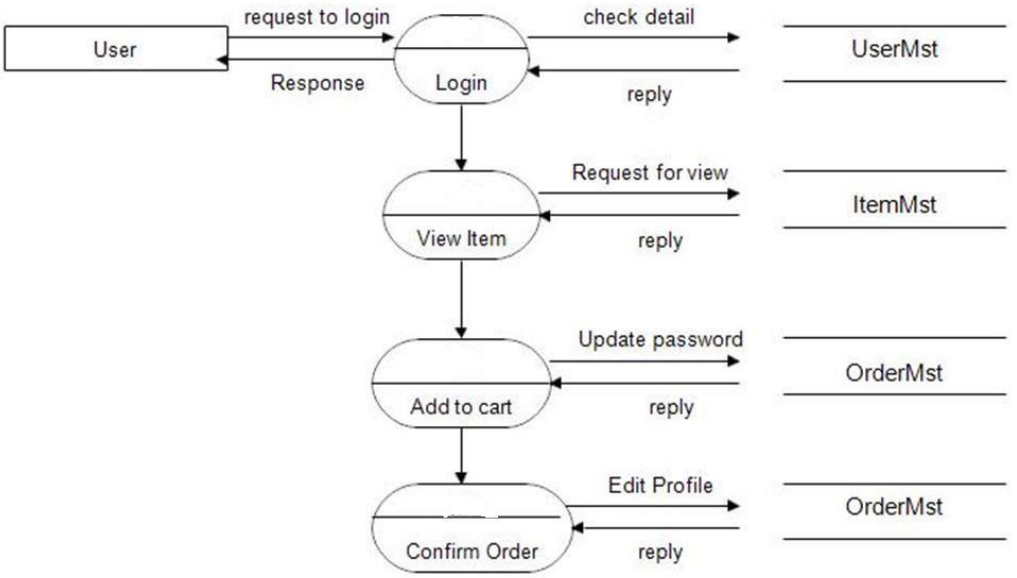
Level 1 User side DFD

The user is all people who operate or visit our website. User is a customer of a website. User can first select product to buy, user must have to register in our system to purchase any item from our website. After registering, user can login to site and buy item



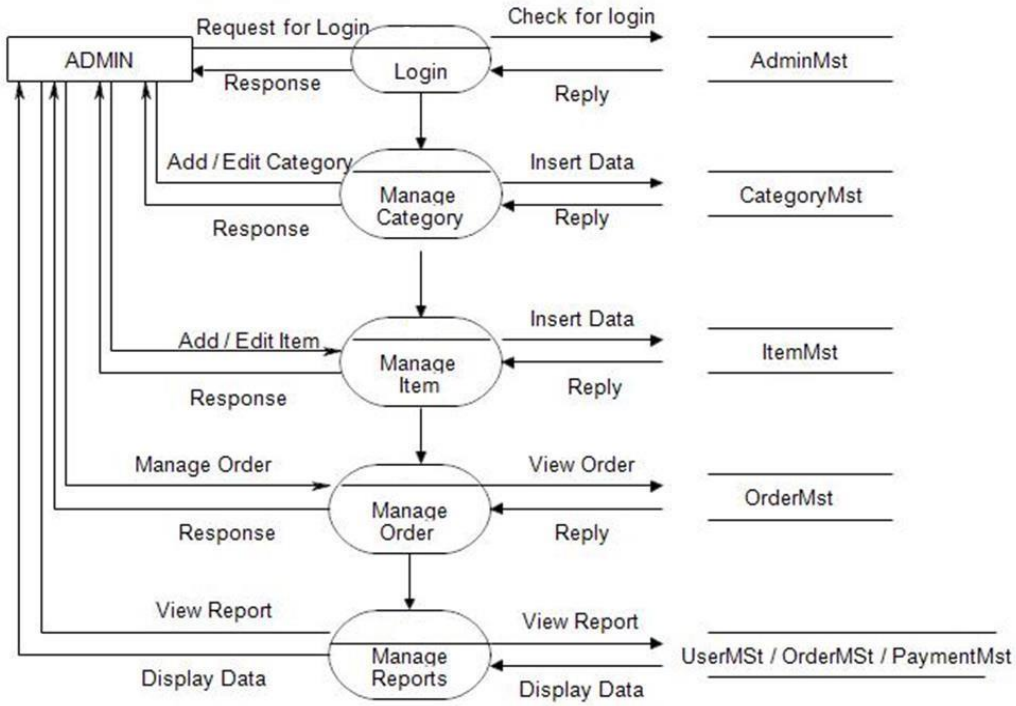
Level 2 User side DFD



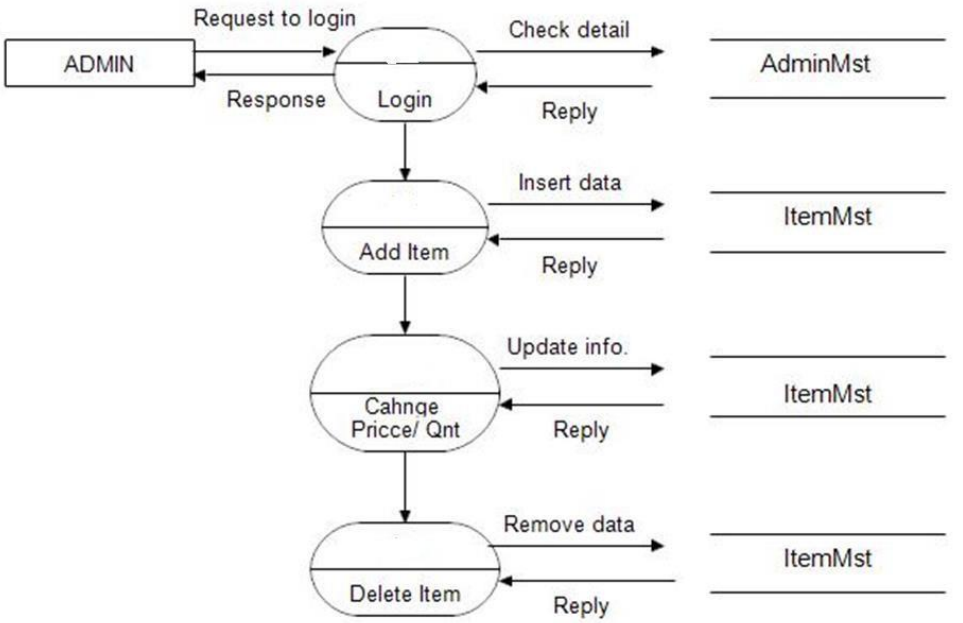


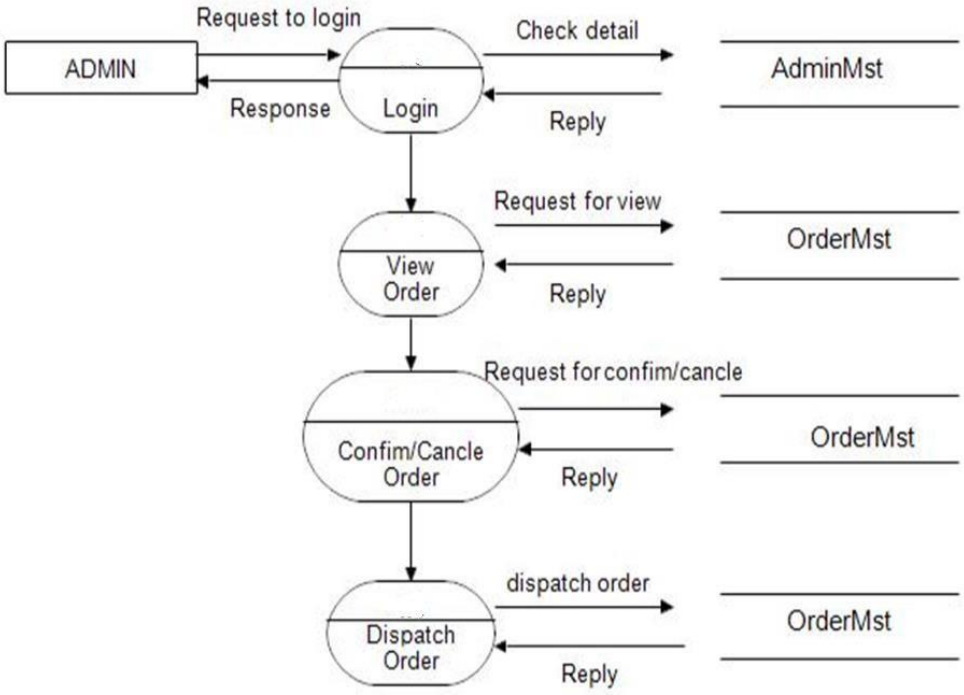
Level 1 Admin side DFD

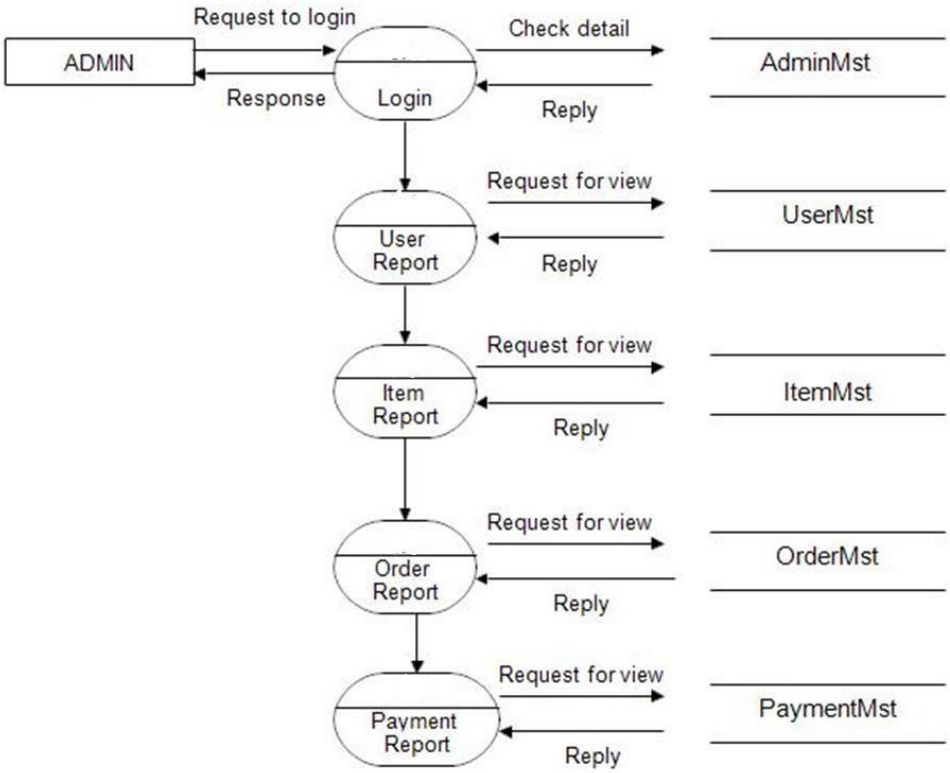
The admin side DFD describe the functionality of Admin, Admin is an owner of the website. Admin can first add category of item and then add items by category wise and admin can manage order and payment detail the admin side DFD describe the functionality of Admin, Admin is an owner of the website. Admin can first add category of item and then add items by category wise and admin can manage order and payment detail.



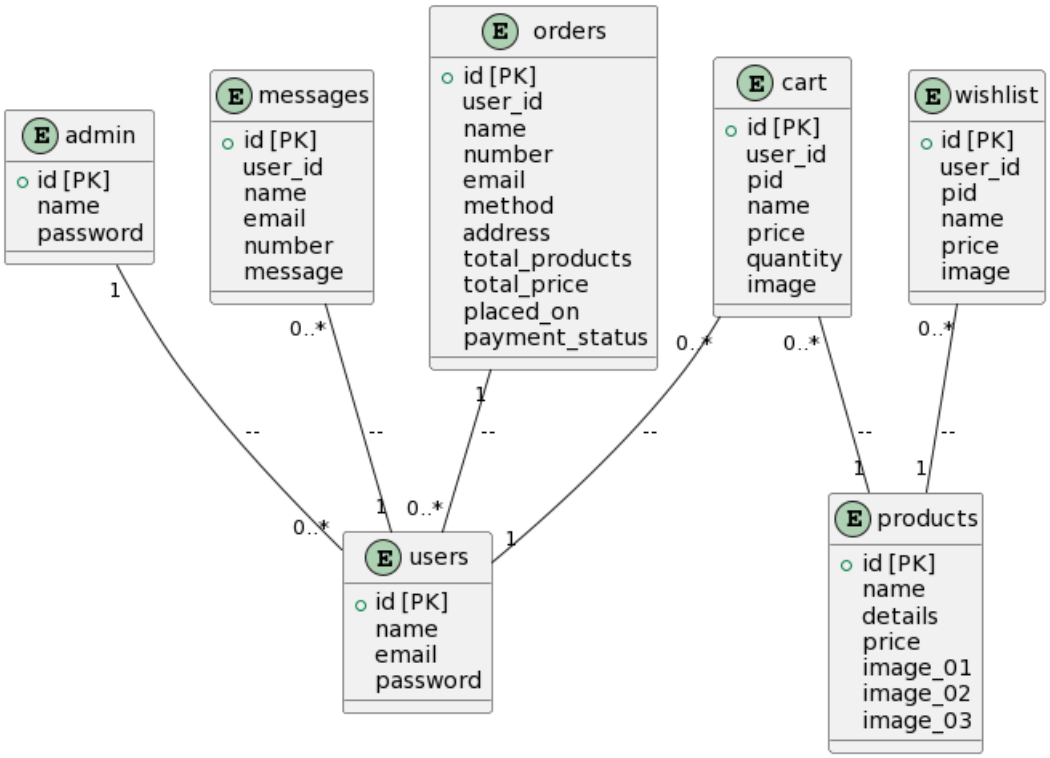
Level 2 Admin side DFD







6. Database Design:



Entities:

Admin: Represents administrators of the system who have privileges to manage users, products, orders, etc.

Cart: Stores information about the items added to the shopping cart by users. Each entry in the cart is associated with a specific user and product.

Messages: Stores messages sent by users. This could be used for customer support or feedback purposes.

Orders: Contains details of orders placed by users, including order items, shipping information, and payment status.

Products: Contains information about the products available for sale, such as name, price, and images.

Users: Represents registered users of the platform, storing their basic information like name, email, and password.

Wishlist: Stores the products that users have added to their wishlist for future reference or purchase.

Attributes:

Each entity has attributes (fields) that describe the properties of that entity. For example, the products entity has attributes like name, details, price, and images.

Attributes are listed along with their data types. For instance, id attributes are marked as primary keys [PK] to indicate uniqueness within their respective tables.

Relationships:

Relationships between entities are represented using lines connecting them.

The cardinality of the relationships is indicated on each end of the line. For example, "1" -- "0..*" means one-to-many relationship where one entity is associated with zero or more entities of another type.

For example, an admin manages users, cart, messages, and orders.

Users can have multiple entries in the cart, messages, and orders tables, indicating that a user can have multiple carts, messages, or orders associated with them.

Products can be associated with multiple entries in the cart and Wishlist tables, indicating that a product can appear in multiple users' carts or Wishlist.

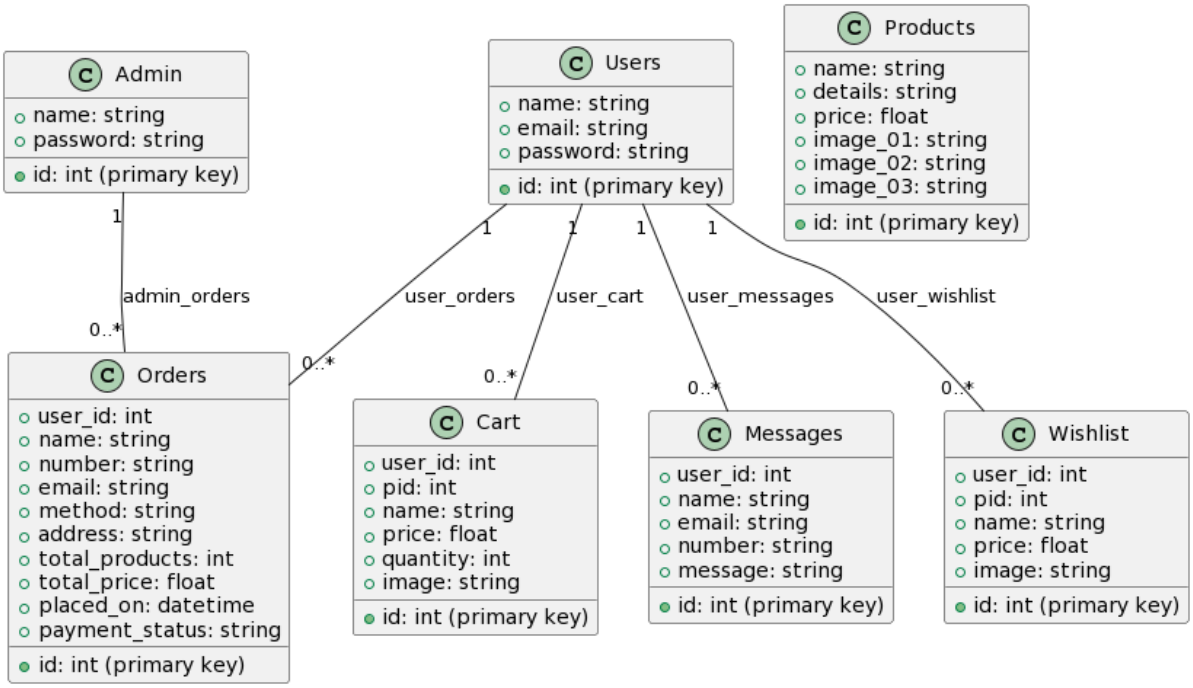
Each entity can have one or more relationships with other entities, defining how they are connected and interact with each other in the database.

Normalization:

This schema follows the principles of normalization to reduce redundancy and improve data integrity.

For example, user-related information is stored in the users table, and orders are linked to users through their user_id, rather than duplicating user information in the orders table.

Class Diagram



Classes: Each table in your database schema is represented as a class in the diagram. For example, Admin, Cart, Messages, Orders, Products, Users, and Wishlist are classes corresponding to the tables in your database (admin, cart, messages, orders, products, users, and Wishlist respectively).

Attributes: Attributes of each class represent the columns in the corresponding database table. For instance, in the admin class, id, name, and password are attributes which map to the columns with the same names in the admin table.

Relationships: Relationships between classes depict the associations between tables in the database.

In the provided diagram, relationships are represented with arrows between classes. The cardinality of the relationship is shown near the end of each arrow.

For instance, the relationship between Admin and Orders indicates that an admin can be associated with zero or more orders (1 to 0..*). This implies that an order can be placed by an admin, but an admin may not necessarily have any associated orders.

Association Names: Association names provide additional information about the relationship between classes. These names describe the nature of the association.

For example, the association name admin_orders between Admin and Orders suggests that it's the orders placed by admins.

Primary Keys: The attributes designated as (primary key) in the class diagram denote the primary key attributes of each table. These are unique identifiers for each record in the table.

Coding:

1. User side:

Home page:

```php
<?php include 'components/user_header.php'; ?>

<div class="home-bg">

<section class="home">

   <div class="swiper home-slider">

   <div class="swiper-wrapper">

      <div class="swiper-slide slide">
         <div class="image">
            <img src="images/home.png" alt="">
         </div>
         <div class="content">
            <span>upto 50% off</span>
            <a href="shop.php" class="btn">shop now</a>
         </div>
      </div>

      <div class="swiper-slide slide">
         <div class="image">
            <img src="images/khadi.png" alt="">
         </div>
         <div class="content">
            <span>upto 50% off</span>
            <a href="shop.php" class="btn">shop now</a>
         </div>
      </div>

      <div class="swiper-slide slide">
         <div class="image">
            <img src="images/khadihome.png" alt="">
         </div>
         <div class="content">
```

```php
<section class="home">
   <div class="swiper-wrapper">
      <div class="swiper-slide slide">
         <div class="image">
            <img src="images/khadihome.png" alt="">
         </div>
         <div class="content">
            <span>upto 50% off</span>
            <a href="shop.php" class="btn">shop now</a>
         </div>
      </div>

   </div>

      <div class="swiper-pagination"></div>

   </div>

</section>

</div>

<section class="category">
   <h1 class="heading">shop by category</h1>
   <div class="swiper category-slider">
      <div class="swiper-wrapper">

         <a href="category.php?category=woman" class="swiper-slide slide">
            <img src="images/women1.jpg" alt="">
            <h3>women</h3>
         </a>

         <a href="category.php?category=men" class="swiper-slide slide">
            <img src="images/menmen.jpg" alt="">
            <h3>men</h3>
         </a>
```

```
<section class="category">
    <div class="swiper category-slider">
        <div class="swiper-wrapper">
            <a href="category.php?category=men" class="swiper-slide slide">
            </a>

            <a href="category.php?category=kids" class="swiper-slide slide">
                <img src="images/kids.jpg" alt="">
                <h3>kids</h3>
            </a>

        </div>
    </div>
</section>



<section class="home-products">

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

   <div class="swiper products-slider">

   <div class="swiper-wrapper">

   <?php
     $select_products = $conn->prepare("SELECT * FROM `products` LIMIT 6");
     $select_products->execute();
     if($select_products->rowCount() > 0){
      while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
   ?>
   <form action="" method="post" class="swiper-slide slide">
      <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
      <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
      <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
      <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">
      <button class="fas fa-heart" type="submit" name="add to wishlist"></button>
```

```
 <section class="home-products">
    <?php
      $select_products->execute();
      if($select_products->rowCount() > 0){
       while($fetch_product = $select_products->fetch(PDO::FETCH_ASSOC)){
    ?>
    <form action="" method="post" class="swiper-slide slide">
       <input type="hidden" name="pid" value="<?= $fetch_product['id']; ?>">
       <input type="hidden" name="name" value="<?= $fetch_product['name']; ?>">
       <input type="hidden" name="price" value="<?= $fetch_product['price']; ?>">
       <input type="hidden" name="image" value="<?= $fetch_product['image_01']; ?>">
       <button class="fas fa-heart" type="submit" name="add_to_wishlist"></button>
       <a href="quick_view.php?pid=<?= $fetch_product['id']; ?>" class="fas fa-eye"></a>
       <img src="uploaded_img/<?= $fetch_product['image_01']; ?>" alt="">
       <div class="name"><?= $fetch_product['name']; ?></div>
       <div class="flex">
          <div class="price"><span>Rs.</span><?= $fetch_product['price']; ?><span>/-</span></div>
          <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;" value="1">
       </div>
       <input type="submit" value="add to cart" class="btn" name="add_to_cart">
    </form>
    <?php
       }
    }else{
       echo '<p class="empty">no products added yet!</p>';
    }
    ?>

    </div>

    <div class="swiper-pagination"></div>

    </div>

 </section>
```

Wishlist:

```
<?php include 'components/user_header.php'; ?>
<section class="products">

   <h3 class="heading">your wishlist</h3>

   <div class="box-container">

   <?php
      $grand_total = 0;
      $select_wishlist = $conn->prepare("SELECT * FROM `wishlist` WHERE user_id = ?");
      $select_wishlist->execute([$user_id]);
      if($select_wishlist->rowCount() > 0){
         while($fetch_wishlist = $select_wishlist->fetch(PDO::FETCH_ASSOC)){
            $grand_total += $fetch_wishlist['price'];
   ?>
   <form action="" method="post" class="box">
      <input type="hidden" name="pid" value="<?= $fetch_wishlist['pid']; ?>">
      <input type="hidden" name="wishlist_id" value="<?= $fetch_wishlist['id']; ?>">
      <input type="hidden" name="name" value="<?= $fetch_wishlist['name']; ?>">
      <input type="hidden" name="price" value="<?= $fetch_wishlist['price']; ?>">
      <input type="hidden" name="image" value="<?= $fetch_wishlist['image']; ?>">
      <a href="quick_view.php?pid=<?= $fetch_wishlist['pid']; ?>" class="fas fa-eye"></a>
      <img src="uploaded_img/<?= $fetch_wishlist['image']; ?>" alt="">
      <div class="name"><?= $fetch_wishlist['name']; ?></div>
      <div class="flex">
         <div class="price">Rs.<?= $fetch_wishlist['price']; ?>/-</div>
         <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;" value="1">
      </div>
      <input type="submit" value="add to cart" class="btn" name="add_to_cart">
      <input type="submit" value="delete item" onclick="return confirm('delete this from wishlist?');" class="delete-btn" name="delete">
   </form>
   <?php
      }
   }else{
      echo '<p class="empty">your wishlist is empty</p>';
   }
```

Cart:

```php
<?php include 'components/user_header.php'; ?>
<section class="products shopping-cart">

   <h3 class="heading">shopping cart</h3>

   <div class="box-container">

   <?php
      $grand_total = 0;
      $select_cart = $conn->prepare("SELECT * FROM `cart` WHERE user_id = ?");
      $select_cart->execute([$user_id]);
      if($select_cart->rowCount() > 0){
         while($fetch_cart = $select_cart->fetch(PDO::FETCH_ASSOC)){
   ?>
   <form action="" method="post" class="box">
      <input type="hidden" name="cart_id" value="<?= $fetch_cart['id']; ?>">
      <a href="quick_view.php?pid=<?= $fetch_cart['pid']; ?>" class="fas fa-eye"></a>
      <img src="uploaded_img/<?= $fetch_cart['image']; ?>" alt="">
      <div class="name"><?= $fetch_cart['name']; ?></div>
      <div class="flex">
         <div class="price">Rs.<?= $fetch_cart['price']; ?>/-</div>
         <input type="number" name="qty" class="qty" min="1" max="99" onkeypress="if(this.value.length == 2) return false;" value="<?= $fetch_cart['quantity']; ?>">
         <button type="submit" class="fas fa-edit" name="update_qty"></button>
      </div>
      <div class="sub-total"> sub total : <span>Rs.<?= $sub_total = ($fetch_cart['price'] * $fetch_cart['quantity']); ?>/-</span> </div>
      <input type="submit" value="delete item" onclick="return confirm('delete this from cart?');" class="delete-btn" name="delete">
   </form>
   <?php
   $grand_total += $sub_total;
      }
   }else{
      echo '<p class="empty">your cart is empty</p>';
   }
   ?>
   </div>
```

2. Admin side:

Dashboard:

```php
<?php include '../components/admin_header.php'; ?>

<section class="dashboard">

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

   <div class="box-container">

      <div class="box">
         <h3>welcome!</h3>
         <p><?= $fetch_profile['name']; ?></p>
         <a href="update_profile.php" class="btn">update profile</a>
      </div>

      <div class="box">
         <?php
            $total_pendings = 0;
            $select_pendings = $conn->prepare("SELECT * FROM `orders` WHERE payment_status = ?");
            $select_pendings->execute(['pending']);
            if($select_pendings->rowCount() > 0){
               while($fetch_pendings = $select_pendings->fetch(PDO::FETCH_ASSOC)){
                  $total_pendings += $fetch_pendings['total_price'];
               }
            }
         ?>
         <h3><span>Rs.</span><?= $total_pendings; ?><span>/-</span></h3>
         <p>total pendings</p>
         <a href="placed_orders.php" class="btn">see orders</a>
      </div>

      <div class="box">
         <?php
            $total_completes = 0;
            $select_completes = $conn->prepare("SELECT * FROM `orders` WHERE payment_status = ?");
            $select_completes->execute(['completed']);
            if($select_completes->rowCount() > 0){
```

```php
<section class="dashboard">
   <div class="box-container">
      <div class="box">
         <?php
            $select_completes = $conn->prepare("SELECT * FROM `orders` WHERE payment_status = ?");
            $select_completes->execute(['completed']);
            if($select_completes->rowCount() > 0){
               while($fetch_completes = $select_completes->fetch(PDO::FETCH_ASSOC)){
                  $total_completes += $fetch_completes['total_price'];
               }
            }
         ?>
         <h3><span>Rs.</span><?= $total_completes; ?><span>/-</span></h3>
         <p>completed orders</p>
         <a href="placed_orders.php" class="btn">see orders</a>
      </div>

      <div class="box">
         <?php
            $select_orders = $conn->prepare("SELECT * FROM `orders`");
            $select_orders->execute();
            $number_of_orders = $select_orders->rowCount()
         ?>
         <h3><?= $number_of_orders; ?></h3>
         <p>orders placed</p>
         <a href="placed_orders.php" class="btn">see orders</a>
      </div>

      <div class="box">
         <?php
            $select_products = $conn->prepare("SELECT * FROM `products`");
            $select_products->execute();
            $number_of_products = $select_products->rowCount()
         ?>
         <h3><?= $number_of_products; ?></h3>
         <p>products added</p>
         <a href="products.php" class="btn">see products</a>
```

Products:

```php
<?php include '../components/admin_header.php'; ?>
<section class="orders">
<h1 class="heading">placed orders</h1>
<div class="box-container">
   <?php
      $select_orders = $conn->prepare("SELECT * FROM `orders`");
      $select_orders->execute();
      if($select_orders->rowCount() > 0){
         while($fetch_orders = $select_orders->fetch(PDO::FETCH_ASSOC)){
   ?>
   <div class="box">
      <p> placed on : <span><?= $fetch_orders['placed_on']; ?></span> </p>
      <p> name : <span><?= $fetch_orders['name']; ?></span> </p>
      <p> number : <span><?= $fetch_orders['number']; ?></span> </p>
      <p> address : <span><?= $fetch_orders['address']; ?></span> </p>
      <p> total products : <span><?= $fetch_orders['total_products']; ?></span> </p>
      <p> total price : <span>Rs.<?= $fetch_orders['total_price']; ?>/-</span> </p>
      <p> payment method : <span><?= $fetch_orders['method']; ?></span> </p>
      <form action="" method="post">
         <input type="hidden" name="order_id" value="<?= $fetch_orders['id']; ?>">
         <select name="payment_status" class="select">
            <option selected disabled><?= $fetch_orders['payment_status']; ?></option>
            <option value="pending">pending</option>
            <option value="completed">completed</option>
         </select>
        <div class="flex-btn">
         <input type="submit" value="update" class="option-btn" name="update_payment">
         <a href="placed_orders.php?delete=<?= $fetch_orders['id']; ?>" class="delete-btn" onclick="return confirm('delete this order?');">delete</a>
        </div>
      </form>
   </div>
   <?php
         }
      }else{
         echo '<p class="empty">no orders placed yet!</p>';
      }
   ?>
```

Testing:

User Side Testing:

User Registration:

Test case 1: Verified that users can register with valid email, password, and other required information.

Test case 2: Verified that users cannot register with invalid or already used email addresses.

Test case 3: Verified that password requirements are enforced (e.g., minimum length, special characters).

Login Functionality:

Test case 4: Verified that registered users can log in with valid credentials.

Test case 5: Verified that users cannot log in with incorrect credentials.

Test case 6: Verified that users can reset their password if forgotten.

Browsing Products:

Test case 7: Verified that users can browse products by categories.

Test case 8: Verified that users can search for products using the search bar.

Test case 9: Verified that product images, descriptions, and prices are displayed correctly.

Adding to Cart:

Test case 10: Verified that users can add products to the shopping cart.

Test case 11: Verified that users can update the quantity of products in the cart.

Test case 12: Verified that users can remove products from the cart.

Checkout Process:

Test case 13: Verified that users can proceed to checkout from the cart.

Test case 14: Verified that users can enter shipping and billing information.

Test case 15: Verified that users can select shipping methods and payment options.

Website Performance:

Test case 16: Verified that the website loads quickly and efficiently.

Test case 17: Verified that the website is responsive across different devices and screen sizes.

Admin Side Testing:

Admin Login:

Test case 18: Verified that admin users can log in with valid credentials.

Test case 19: Verified that admin users cannot log in with incorrect credentials.

Test case 20: Verified that admin users can reset their password if forgotten.

Product Management:

Test case 21: Verified that admins can add new products to the inventory.

Test case 22: Verified that admins can edit existing product details.

Test case 23: Verified that admins can delete products from the inventory.

Order Management:

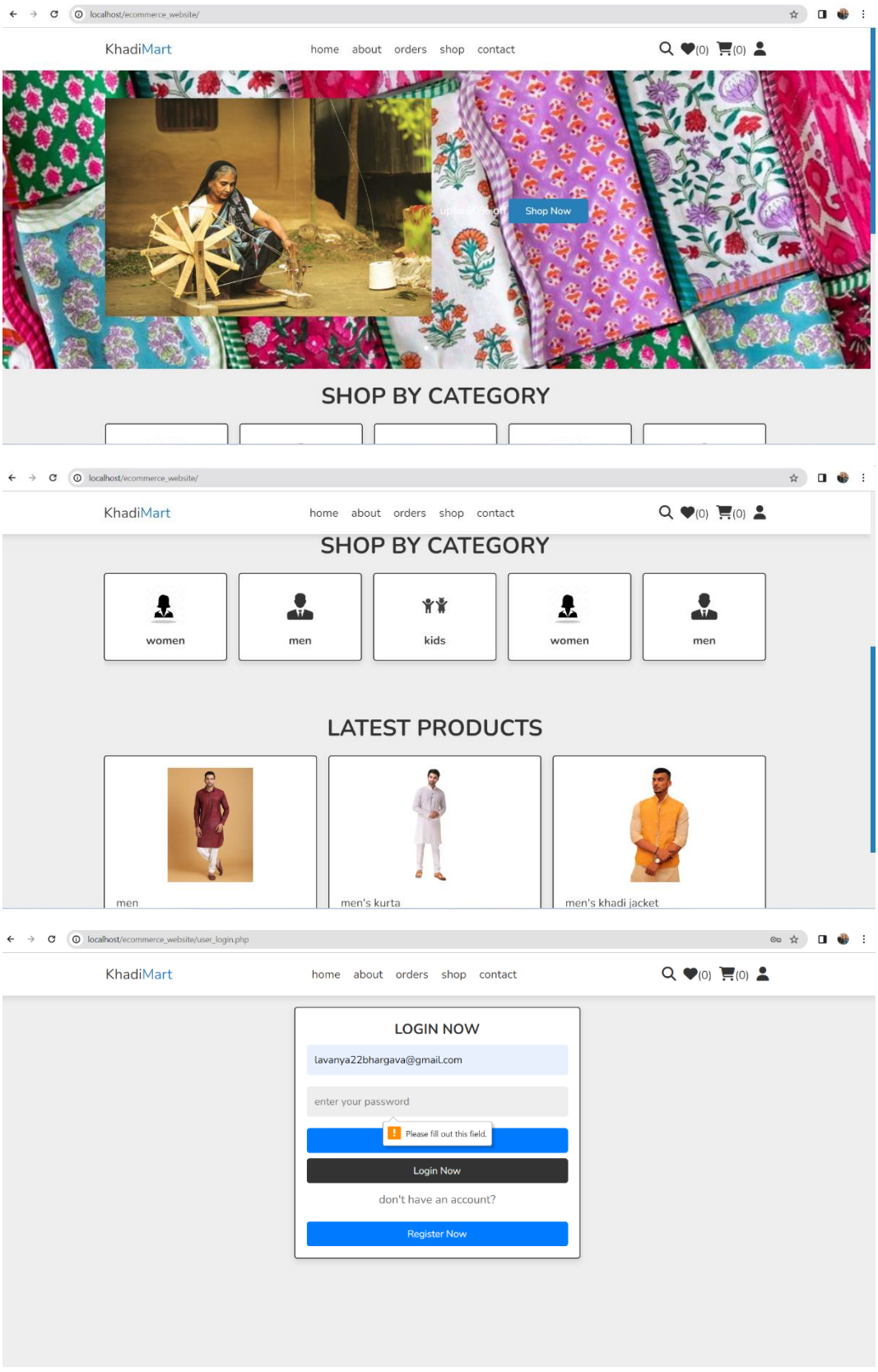
Test case 24: Verified that admins can view a list of all orders placed.

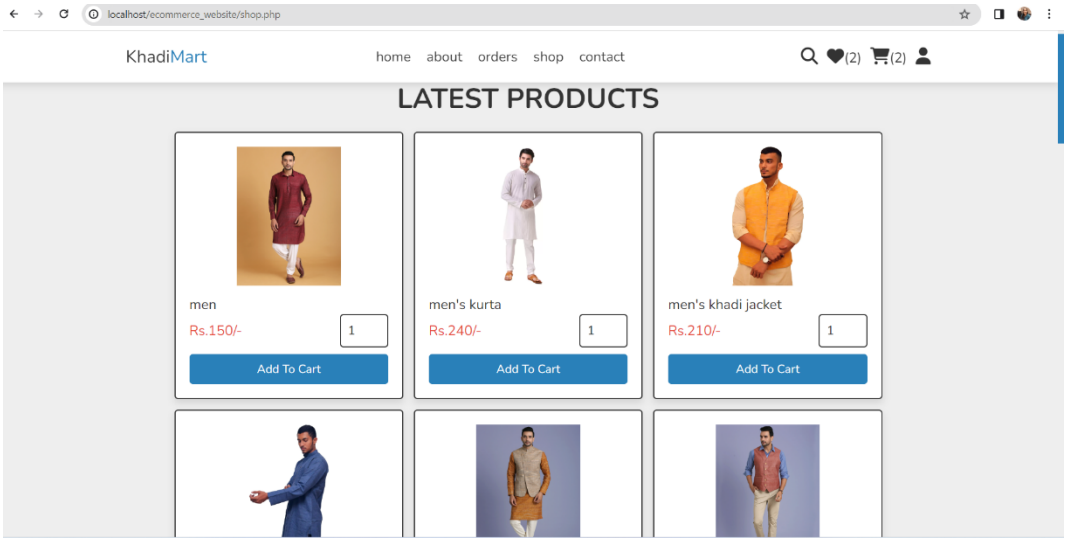
Test case 25: Verified that admins can update the status of orders (e.g., processing, shipped, delivered).

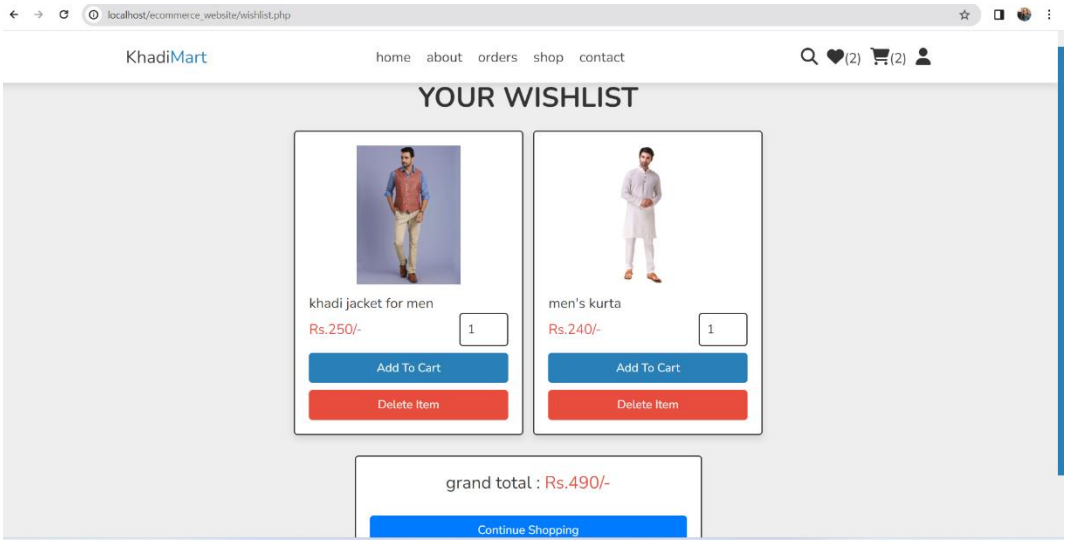
Test case 26: Verified that admins can view order details including customer information and purchased items.

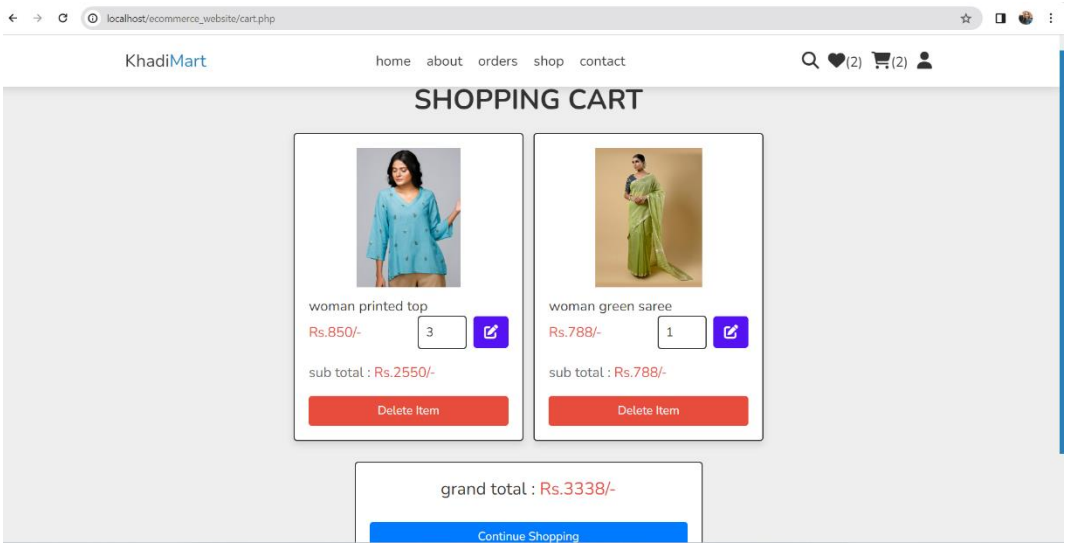
User Interfaces:

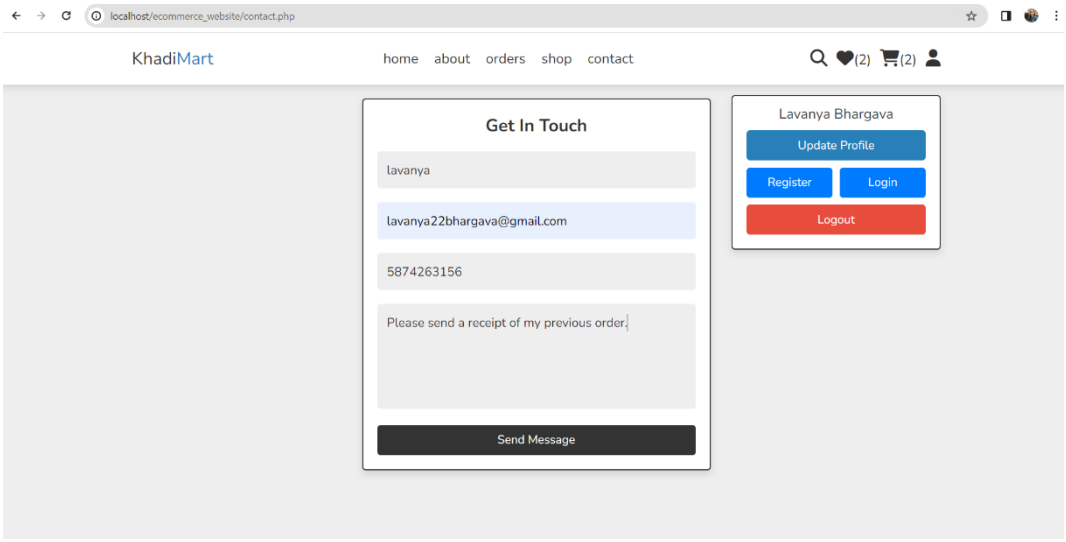
User side:

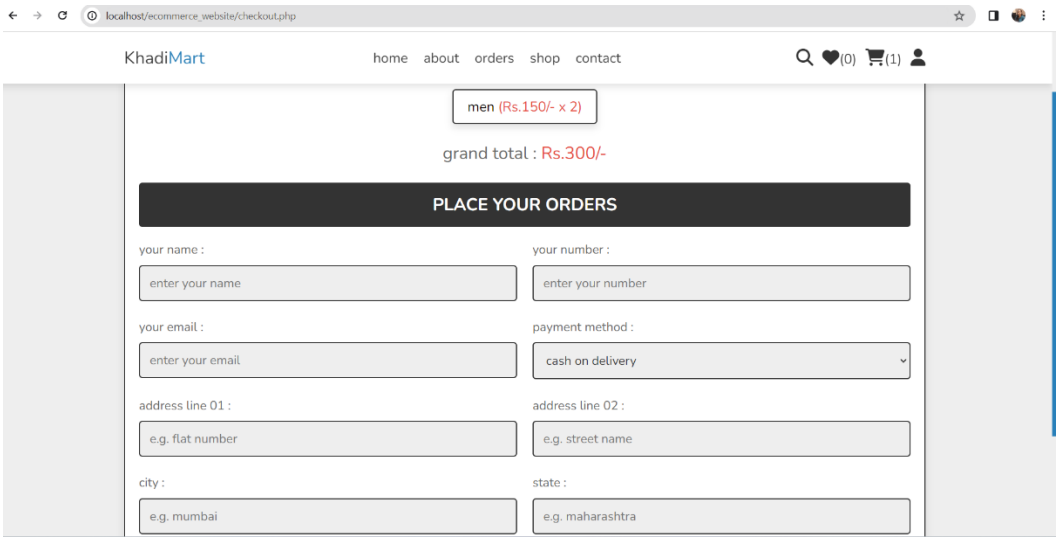



localhost/ecommerce_website/shop.php
KhadiMart
home about orders shop contact
(2) (2)
LATEST PRODUCTS
men
Rs.150/-
1
Add To Cart
men's kurta
Rs.240/-
1
Add To Cart
men's khadi jacket
Rs.210/-
1
Add To Cart

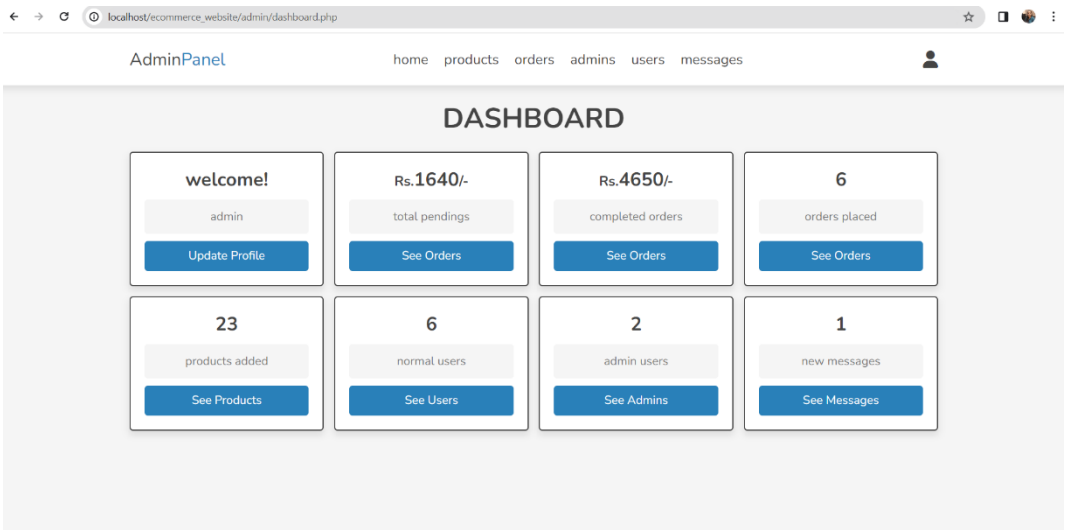

localhost/ecommerce_website/wishlist.php
KhadiMart
home about orders shop contact
(2) (2)
YOUR WISHLIST
khadi jacket for men
Rs.250/-
1
Add To Cart
Delete Item
men's kurta
Rs.240/-
1
Add To Cart
Delete Item
grand total : Rs.490/-
Continue Shopping

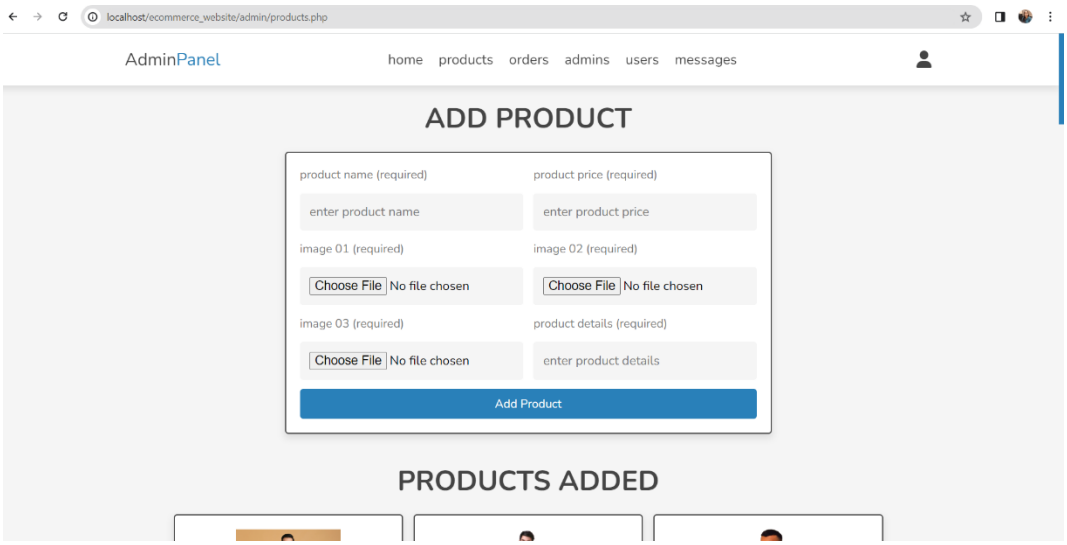

localhost/ecommerce_website/cart.php
KhadiMart
home about orders shop contact
(2) (2)
SHOPPING CART
woman printed top
Rs.850/-
3
sub total : Rs.2550/-
Delete Item
woman green saree
Rs.788/-
1
sub total : Rs.788/-
Delete Item
grand total : Rs.3338/-
Continue Shopping

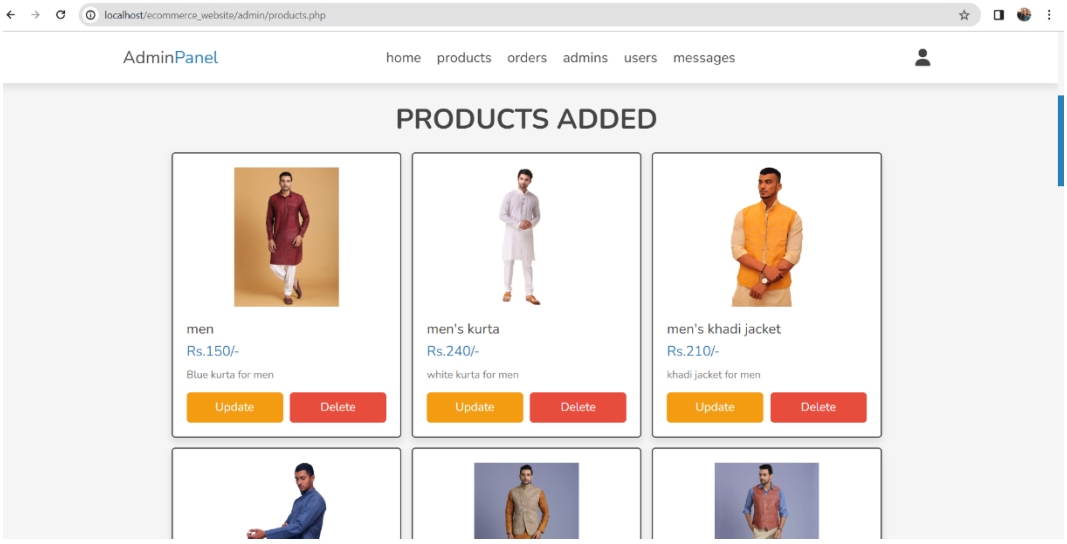

localhost/ecommerce_website/contact.php
KhadiMart
home about orders shop contact
(2) (2)
Get In Touch
lavanya
lavanya22bhargava@gmail.com
5874263156
Please send a receipt of my previous order.
Send Message
Lavanya Bhargava
Update Profile
Register
Login
Logout

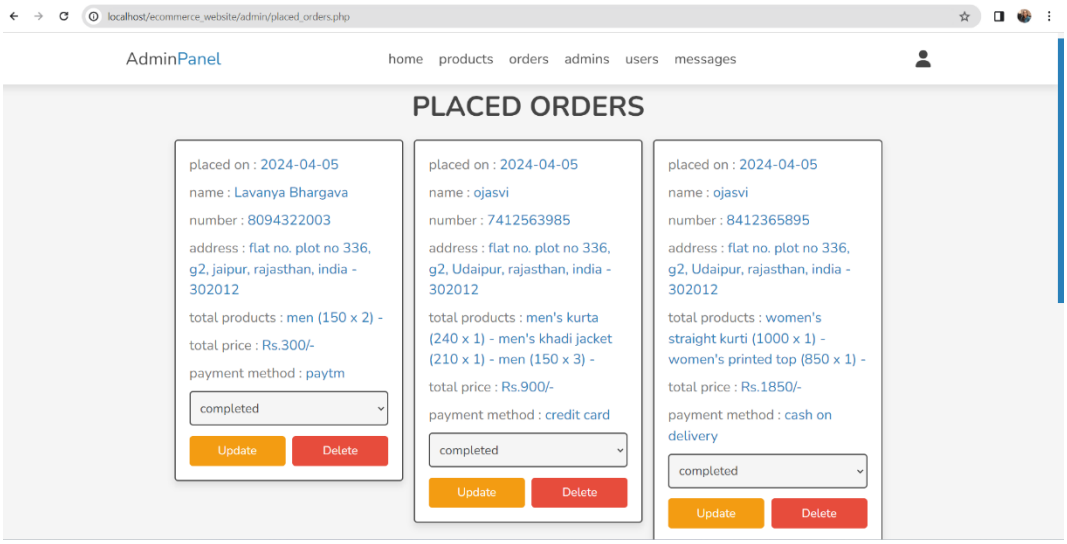

localhost/ecommerce_website/checkout.php
KhadiMart
home about orders shop contact
(0) (1)
men (Rs.150/- x 2)
grand total : Rs.300/-
PLACE YOUR ORDERS
your name :
enter your name
your number :
enter your number
your email :
enter your email
payment method :
cash on delivery
address line 01 :
e.g. flat number
address line 02 :
e.g. street name
city :
e.g. mumbai
state :
e.g. maharashtra

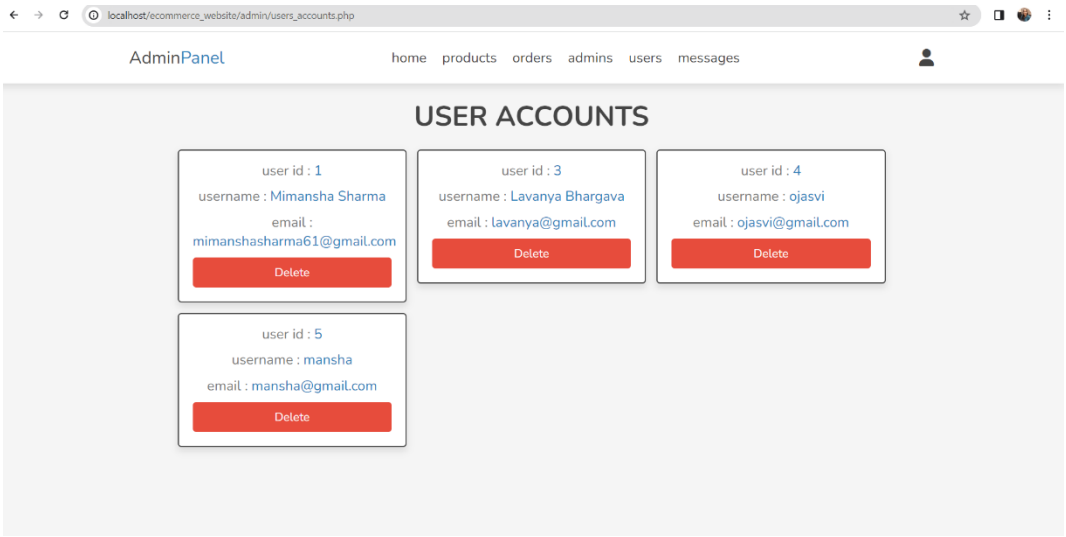
Admin side:


localhost/ecommerce_website/admin/dashboard.php
AdminPanel
home products orders admins users messages
DASHBOARD
welcome!
admin
Update Profile
Rs.1640/-
total pendings
See Orders
Rs.4650/-
completed orders
See Orders
6
orders placed
See Orders
23
products added
See Products
6
normal users
See Users
2
admin users
See Admins
1
new messages
See Messages


localhost/ecommerce_website/admin/products.php
AdminPanel
home products orders admins users messages
ADD PRODUCT
product name (required)
enter product name
product price (required)
enter product price
image 01 (required)
Choose File No file chosen
image 02 (required)
Choose File No file chosen
image 03 (required)
Choose File No file chosen
product details (required)
enter product details
Add Product
PRODUCTS ADDED


localhost/ecommerce_website/admin/products.php
AdminPanel
home products orders admins users messages
PRODUCTS ADDED
men
Rs.150/-
Blue kurta for men
Update
Delete
men's kurta
Rs.240/-
white kurta for men
Update
Delete
men's khadi jacket
Rs.210/-
khadi jacket for men
Update
Delete


localhost/ecommerce_website/admin/placed_orders.php
AdminPanel
home products orders admins users messages
PLACED ORDERS
placed on : 2024-04-05
name : Lavanya Bhargava
number : 8094322003
address : flat no. plot no 336, g2, jaipur, rajasthan, india - 302012
total products : men (150 x 2) -
total price : Rs.300/-
payment method : paytm
completed
Update
Delete
placed on : 2024-04-05
name : ojasvi
number : 7412563985
address : flat no. plot no 336, g2, Udaipur, rajasthan, india - 302012
total products : men's kurta (240 x 1) - men's khadi jacket (210 x 1) - men (150 x 3) -
total price : Rs.900/-
payment method : credit card
completed
Update
Delete
placed on : 2024-04-05
name : ojasvi
number : 8412365895
address : flat no. plot no 336, g2, Udaipur, rajasthan, india - 302012
total products : women's straight kurti (1000 x 1) - women's printed top (850 x 1) -
total price : Rs.1850/-
payment method : cash on delivery
completed
Update
Delete


localhost/ecommerce_website/admin/users_accounts.php
AdminPanel
home products orders admins users messages
USER ACCOUNTS
user id : 1
username : Mimansha Sharma
email : mimanshasharma61@gmail.com
Delete
user id : 3
username : Lavanya Bhargava
email : lavanya@gmail.com
Delete
user id : 4
username : ojasvi
email : ojasvi@gmail.com
Delete
user id : 5
username : mansha
email : mansha@gmail.com
Delete

References:

https://opus.govst.edu/cgi/viewcontent.cgi?article=1079&context=capstones

https://creately.com/diagram/example/hzvi5flu/er-diagram-for-online-shoppingsystem

https://www.slideshare.net/rabartkurrey/se-dfdr-ep

https://www.lucidchart.com/pages/uml-component-diagram

https://www.geeksforgeeks.org/types-software-testing/