



CAN 302 Coursework 2

E-commerce Solution to New Energy Batteries for Automobiles

Team Member	Student ID
Yeheng Ma	1822070
Xinyun Fan	1823923
Yingshuang Zhang	1825246
Demi Li	1825252
Jizheng Chen	1825254

Abstract

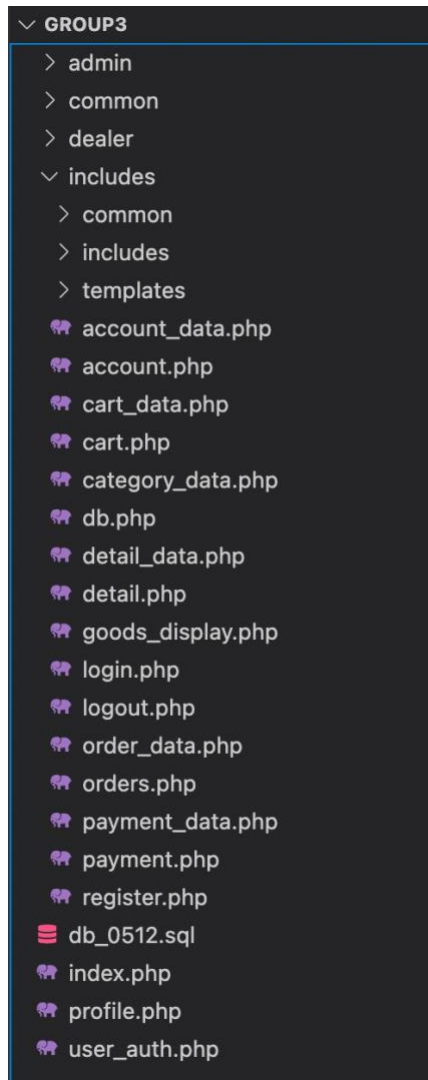
Following the previous report, this report mainly aims to provide specific solutions for the e-commerce sales model of new energy batteries. It has been clarified in previous reports that the online sales model of new energy batteries has huge potential value, and e-commerce aims to conduct transaction activities and provide other services through the Internet or electronic transactions. In response to this, this report launched the online sales web version of new energy batteries. The solution includes the triple realization of the user side, admin side and dealer side of the new energy battery, that is, the web page allows customers, administrators and dealers to register, log in, sell, purchase, manage orders, manage products, payment and so on for new energy batteries. Our team is not satisfied with just making examples of web pages, so within the limited time and energy, our team members did their best to realize the various functions and logical frameworks of the web pages to the greatest extent.

Contents

<i>Abstract</i>	<i>ii</i>
<i>Contents</i>	<i>iii</i>
1 Solution architecture	1
2 Change to function list	2
3 Function implementation.....	4
3.1. Login.....	4
3.2. Register.....	6
3.3. Edit account.....	8
3.4. Browse merchandise	10
3.5. Search for merchandise.....	13
3.6. Browse detailed information.....	14
3.7. Manage cart.....	15
3.8. Purchase and payment.....	18
3.9. Refund	21
3.10. Add payment method.....	22
3.11. Browse orders.....	23
3.12. Talk to customer services	25
3.13. Browse inventory.....	26
3.14. Manage product status.....	27
3.15. Manage orders.....	28
3.16. Browse statistics.....	29
3.17. Browse dealer data	30
4 Data structure	31
4.1. Database structure	32
4.2. Table structure.....	32

1 Solution architecture

This solution is built with user-interface deployed on web server and database deployed on MySQL server. The database structure will be introduced in part 4. The user-interface framework in as following:



2 Change to function list

This part shows the changes made to the function list compared with the previous report and the corresponding reasons and explanations. For specific e-commerce solutions and detailed explanations, this report is based on the whole function list. For details, please see part 3. It should be noted that in this report, the modification we made to the previous function list is deletion and simplification, which does not mean that our interface functions are reduced a very large part. In fact, we have mainly deleted some secondary functions, which are mainly due to technical problems or unnecessary. In the previous report, our team members have designed the function list in detail, so we don't need to add more functions here.

Function	Change	User story (Previous reason for including)	Updated reason
Role: Customer			
Browse history	Deleted	<ul style="list-style-type: none">As a consumer, I want to know what I browsed just now.Specifically, as a customer, after browsing many products, I want to find the products I browsed before, but due to the large number of products, the original path is no longer available, so I hope to find the target product through the history.	The current webpage version is an example stage, so we only add a small number of products as examples. Due to the small number of products, all products can be easily found on the homepage, and users do not need to find their target product through history, so there is no need to add the browsing history function.
Talk to	Simplified	As a consumer, I want to talk	Originally, we planned to

customer services		to customer services.	design a customer service communication interface like TaoBao, but according to the existing knowledge, we have no way to design a chat-style interface. Therefore, after consulting the teacher, we chose to only add the contact number of the relevant supplier at the bottom of the homepage to provide communication services for customers.
Role: Dealer			
Edit product information	Deleted	As a dealer, I want to edit product information.	Dealers can directly operate the database, so there is no need to have a graphical interface on the web page to realize this function.
Manage orders	Simplified	As a dealer, I want to deal with orders from consumers, including accepting payment, deciding whether to refund, and assigning installation services.	It is out of the scope of this project to design an interface for dealers to accept payment and talk to third-party services. To let users know whether their refund request is processed would be good.
Customer communication	Deleted	As a dealer, I want to talk to my customer if required.	See 'talk to customer services' for details. Due to the change of the

			communication interface of the client, there is no need to provide the communication interface for dealer.
Inner communication	Deleted	As a dealer, I want to talk to my administrator if required.	This is a secondary function. It has been explained before that we cannot design the communication interface at this stage, so we can only delete this function. At the same time, it should be clear that our solution is mainly for consumers, and the deletion of this function does not affect normal use.
Role: Administrator			
Inner communication	Deleted	As a manager, I want to talk to my dealers if required.	See 'inner communication' in 'Role: Dealer' part for detail.

3 Function implementation

3.1. Login

Function	User story	Requirement	Implement method
----------	------------	-------------	------------------

Login	As a returning user, I want to login to the website to access the service or information.	<ul style="list-style-type: none"> The system must separate users into consumers, dealers, and the administrator. The system must allow users to login to the website by entering their e-mail address and password. 	PHP accepts the user login information, and then matches and compares it with the database information. If the matching is successful, the login is successful, otherwise the login fails.
-------	---	--	--

PHP file: includes/login.php

Key code:

```
includes > login.php
<?php
if(isset($_POST['login'])){
    $user = $_POST['user'];
    $password = $_POST['password'];
    $sql = "SELECT * FROM user WHERE name='$user' AND password='$password'";
    if ($result = mysqli_query($con, $sql))
    {
        $row = mysqli_fetch_array($result);
        $count = mysqli_num_rows($result);
        if($count==1)
        {
            $_SESSION['user'] = $row['name'];
            $_SESSION['id'] = $row['id'];

            echo "<script>window.open('index.php','_self')</script>";
        }
        else{
            echo "<script>alert('username or Password is Wrong!')</script>";
            echo "<script>window.location.replace('user_auth.php?section=login','_self')</script>";
        }
    }
}
?>
```



```

<!--banner-->
<div class="banner-top">
  <div class="container">
    <h1>Login</h1>
    <em></em>
    <h2><a href="index.php">Go to Home</a><label>||</label>Login</a></h2>
  </div>
</div>
<!-- login-->
<div class="container">
  <div class="login">
    <form action="" method="post">
      <div class="col-md-6 login-do">
        <div class="login-mail">
          <input type="text" name="user" placeholder="username" required="">
          <i class="glyphicon glyphicon-envelope"></i>
        </div>
        <div class="login-mail">
          <input type="password" name="password" placeholder="password" required="">
          <i class="glyphicon glyphicon-lock"></i>
        </div>
        <label class="hvr-skew-backward">
          <button type="submit" name="login" class=" hvr-skew-backward">Login</button>
        </label>
      </div>
      <div class="col-md-6 login-right">
        <h3>Completely Free Account</h3>
        <p>Haven't got an account?</p>
        <a href="user_auth.php?section=register" class=" hvr-skew-backward">Signup</a>
      </div>
    </form>
  </div>
</div>
<!--// login-->

```

3.2. Register

Function	User story	Requirement	Implement method
Sign up	As a new user, I want to register to get a new account.	<ul style="list-style-type: none"> The system must ask the user to claim to be consumers, dealers or the administrator. The system must allow users to create an account by setting a password and 	PHP accepts the information entered by the user, including username, password, email, phone and address, and then inserts the information into the database.

		verifying an email address.	
--	--	--------------------------------	--

PHP file: includes/register.php

Key code:

```
includes > register.php
<?php
if(isset($_POST['register']))
{
    $user = $_POST['user'];
    $phone_number = $_POST['phone_number'];
    $mail = $_POST['mail'];
    $password = $_POST['password'];
    $repassword = $_POST['repassword'];
    $address = $_POST['address'];
    if($user&&$phone_number&&$mail&&$password&&$repassword&&$address&&$password==$repassword){
        echo "<script>console.log('data post success')</script>";
        $sql = "INSERT INTO 'user' ('name', 'phone_number', 'mail', 'password', 'address') VALUES ('$user','$phone_number','$mail','$password','$address') " ;
        $check = mysqli_query($con,$sql);
        if ($check){
            echo "<script>alert('Successfully registered! Login again please')</script>";
            echo "<script>>window.location.replace('user_auth.php?section=login','_self')</script>";
        }
        else{
            echo "<script>alert('Something error')</script>";
            echo "<script>>window.location.replace('user_auth.php?section=register','_self')</script>";
        }
    }
    else{
        echo "<script>alert('data lost or the password is not consistent')</script>";
    }
}
```

```

<!--banner-->
<div class="banner-top">
  <div class="container">
    <h1>Register</h1>
    <em></em>
    <h2><a href="index.php">Go to Home</a><label>||</label>Register</a></h2>
  </div>
</div>
<!--register-->
<div class="container">
  <div class="login">
    <div class="col-md-6 login-do">
      <form action="" method="post">
        <div class="login-mail">
          <input type="text" name = "user" placeholder="username" required="">
          <i class="glyphicon glyphicon-user"></i>
        </div>
        <div class="login-mail">
          <input type="text" name = "phone_number" placeholder="phone number" required="">
          <i class="glyphicon glyphicon-phone"></i>
        </div>
        <div class="login-mail">
          <input type="text" name = "mail" placeholder="email" required="">
          <i class="glyphicon glyphicon-envelope"></i>
        </div>
        <div class="login-mail">
          <input type="password" name = "password" placeholder="password" required="">
          <i class="glyphicon glyphicon-lock"></i>
        </div>
        <div class="login-mail">
          <input type="password" name = "repassword" placeholder="confirm password" required="">
          <i class="glyphicon glyphicon-lock"></i>
        </div>
        <div class="login-mail">
          <input type="text" name = "address" placeholder="address" required="">
          <i class="glyphicon glyphicon-envelope"></i>
        </div>

        <label class="hvr-skew-backward">
          <button class="hvr-skew-backward" name = "register" id = "register" >register</button>
        </label>
      </form>
    </div>

    <div class="col-md-6 login-right">
      <h3>Free Account</h3>
      <p>Already have an account?</p>
      <a href="user_auth.php?section=login" class="hvr-skew-backward">Login</a>
    </div>

    <div class="clearfix"></div>
  </div>
</div>
<!--//register-->

```

3.3. Edit account

Function	User story	Requirement	Implement method
Edit account	As a returning user, I want to change my email address, password.	<ul style="list-style-type: none"> The system must allow users to change email address, and password. The system must require verifying identity by 	PHP accepts the user's modified personal information and updates the user information in database.

		verification code before changes are made.	
--	--	--	--

PHP file: includes/account.php and includes/account_data.php

Key code:

```
<?php include("account_data.php")?>
<div class="col-md-10">
  <div class="check-out">
    <div class="bs-example4" data-example-id="simple-responsive-table">
      <div class="table-responsive">
        <table class="table-heading simpleCart_shelfItem">
          <?php
            if(isset($_POST['edit'])){
              echo "$information_edit";
            }elseif(isset($_POST['update'])){
              $id = $_POST['id'];
              $name = $_POST['name'];
              $phone_number = $_POST['phone_number'];
              $mail = $_POST['mail'];
              $password = $_POST['password'];
              $address = $_POST['address'];
              echo "<script>console.log('data input received')</script>";
              $update = "UPDATE `user` SET `name`='$name', `phone_number`='$phone_number',
                `mail`='$mail', `password`='$password', `address`='$address' WHERE id=$id ";
              $check = mysql_query($con,$update);
              if ($check) {
                echo "<script>window.location.replace('profile.php?section=account','_self')</script>";
                echo "<script>console.log('Profile has been Updated.')</script>";
              }
              else{
                echo "<script>alert('Update failed')</script>";
              }
            }
          }
          else{
            echo "$person_information";
          }
        ?>
      </table>
    </div>
  </div>
</div>
</div>
</div>
```

```

<?php
if(isset($_SESSION['user'])) {
    $uid = $_SESSION['id'];
    $sql = "SELECT * FROM user WHERE id = '$uid'";
    $result = mysqli_query($con, $sql);
    $row = mysqli_fetch_array($result);
    //account information
    $person_information = '
    <form method="post">
        <button type="submit" class="item_add hvr-skew-backward" name="edit">EDIT</button>
    </form>
    <form>
        <tr class="cart-header">
            <td>User name</td>
            <td class="item_price">'. $row['name'] .'</td>
            <td class="add-check"></td>
        </tr>
        <tr class="cart-header1">
            <td>phone number</td>
            <td class="item_price">'. $row['phone_number'] .'</td>
            <td class="add-check"></td>
        </tr>
        <tr class="cart-header2">
            <td>email address</td>
            <td class="item_price">'. $row['mail'] .'</td>
            <td class="add-check"></td>
        </tr>
        <tr class="cart-header2">
            <td>password</td>
            <td class="item_price">*****</td>
            <td class="add-check"></td>
        </tr>
        <tr class="cart-header2">
            <td>Address</td>
            <td class="item_price">'. $row['address'] .'</td>
            <td class="add-check"></td>
        </tr>
    </form>';
}

```

```

//edit account information
$information_edit = '
<form method="post">
    <button type="submit" class="item_add hvr-skew-backward" name="update">Update</button>
    <tr class="cart-header">
        <input type="hidden" name = "id" value='.$row["id"].'></input>
        <td>User name</td>
        <td class="item_price">'. $row['name'] .'</td>
        <td class="add-check"><input name = "name" value='.$row["name"].'></input></td>
    </tr>
    <tr class="cart-header1">
        <td>phone number</td>
        <td class="item_price">'. $row['phone_number'] .'</td>
        <td class="add-check"><input name = "phone_number" value='.$row["phone_number"].'></input></td>
    </tr>
    <tr class="cart-header2">
        <td>email address</td>
        <td class="item_price">'. $row['mail'] .'</td>
        <td class="add-check"><input name = "mail" value='.$row["mail"].'></input></td>
    </tr>
    <tr class="cart-header2">
        <td>password</td>
        <td class="item_price">*****</td>
        <td class="add-check"><input name = "password" value='.$row["password"].'></input></td>
    </tr>
    <tr class="cart-header2">
        <td>Address</td>
        <td class="item_price">'. $row['address'] .'</td>
        <td class="add-check"><input name = "address" value='.$row["address"].'></input></td>
    </tr>
</form>';

} else {
    $person_information = "";
}
?>

```

3.4. Browse merchandise

Function	User story	Requirement	Implement method
Role: Consumer			
Browse merchandise	As a consumer, I want to browse available products on the web page.	The system must layout available products with abstraction view	Read the product information from the database and present it on the web page.

PHP file: index.php and includes/goods_display.php

Key code:

```
<?php
$goods_list = '<div class="mid-popular">';

if(isset($_POST['search'])){
    $input = $_POST['input'];
    $sql = "SELECT g.*, c.name as cname FROM goods g INNER JOIN category c ON g.cid=c.id WHERE g.name LIKE '%" . $input . "%'";
    $cat = 'Result for: ' . $input;
    if ($result = mysqli_query($con, $sql)) {
        $count = mysqli_num_rows($result);
        if($count>0){
            while ($row = mysqli_fetch_array($result)) {
                $goods_list .= '<div class="col-md-3 item-grid simpleCart_shelfItem">
                    <div class="mid-pop">
                        <div class="pro-img">
                            
                            <div class="zoom-icon">
                                <a class="picture" href="includes/detail.php?gid='. $row['gid'] .'"
                                rel="title" class="b-link-stripe b-animate-go thickbox"><i class="glyphicon glyphicon-search icon"></i></a>
                                <a href="index.php?cid='. $row['cid'] .'"><i class="glyphicon glyphicon-menu-right icon"></i></a>
                            </div>
                        </div>
                    </div>
                    <div class="mid-1">
                        <div class="women">
                            <div class="women-top"><span>' . $row['cname'] . '</span>
                                <h6><a>' . $row['name'] . '</a></h6>
                            </div>
                            <div class="img item_add">
                                <form method="post">
                                    <button name="cart_add"></button>
                                    <input type="hidden" name="gid" value=" ' . $row['gid'] . ' "></input>
                                </form>
                            </div>
                            <div class="clearfix">
                                </div>
                        </div>
                        <div class="mid-2">
                            <p><label>V' . $row['original_price'] . '</label><em class="item_price">V' . $row['present_price'] . '</em></p>
                        <div class="block">
                            </div>
                        </div>
                    </div>
                </div>
            }
        }
    }
}
```



```

        <div class="mid-2">
            <p><label>V' . $row['original_price'] . '</label><em class="item_price">V' . $row['present_price'] . '</em></p>
            <div class="block">

        </div>
        <div class="clearfix">

    </div>
</div>
</div>
</div>
</div>';

$goods_list .= '<div class="clearfix"></div></div>';
}
}
?>

```

3.5. Search for merchandise

Function	User story	Requirement	Implement method
Role: Consumer			
Search for merchandise	As a consumer, I want to search for what I want.	The system must allow consumers to search for products by entering keywords of product name or car model.	PHP accepts the user information as the filter condition, and then selects the qualified information from the database to display on the page.

PHP file: index.php and includes/goods_display.php

Key code:


```

if(isset($_POST['search'])){
    $input = $_POST['input'];
    $sql = "SELECT g.*, c.name as cname FROM goods g INNER JOIN category c ON g.cid=c.id WHERE g.name LIKE '%$input%'";
    $cat = 'Result for: '. $input;
    if ($result = mysqli_query($con, $sql)) {
        $count = mysqli_num_rows($result);
        if($count>0){
            while ($row = mysqli_fetch_array($result)) {
                $goods_list .= '<div class="col-md-3 item-grid simpleCart_shelfItem">
                    <div class="mid-pop">
                        <div class="pro-img">
                            
                            <div class="zoom-icon">
                                <a class="picture" href="includes/detail.php?gid='.$row['gid'].'" .'"
                                    rel="title" class="b-link-stripe b-animate-go thickbox"><i class="glyphicon glyphicon-search icon "></i></a>
                                <a href="index.php?cid='.$row['cid'].'"><i class="glyphicon glyphicon-menu-right icon"></i></a>
                            </div>
                        </div>
                    <div class="mid-1">
                        <div class="women">
                            <div class="women-top"><span>' . $row['cname'] .'</span>
                                <h6><a>' . $row['name'] .'</a></h6>
                            </div>
                            <div class="img item_add">
                                <form method="post">
                                    <button name="cart_add"></button>
                                    <input type="hidden" name="gid" value= '.$row['gid'].'></input>
                                </form>
                            </div>
                            <div class="clearfix">
                                </div>
                            </div>
                        <div class="mid-2">
                            <p><label>¥' . $row['original_price'] .'</label><em class="item_price">¥' . $row['present_price'] .'</em></p>
                            <div class="block">
                                </div>
                            <div class="clearfix">
                                </div>
                        </div>
                    </div>
                </div>
            }
        }
    }
}

```

3.6. Browse detailed information

Function	User story	Requirement	Implement method
Role: Consumer			
Browse detailed information	As a consumer, I want to browse detailed information like product parameters on the web page.	The system must allow consumers to access detailed information of products by clicking the button 'details'	PHP identifies the hyperlink of the button, obtains the keyword index of commodity information through the hyperlink, finds out the corresponding commodity in the database through the keyword index and displays the corresponding information.

PHP file: includes/detail_data.php, includes/detail.php and includes/goods_display.php

Key code:

```

!)-Detailed information of goods-->
<?php include("templates/temp_head.php")?>
<?php include("templates/temp_body_header.php")?>
<?php include("detail_data.php")?>
<!--banner-->
<div class="banner-top">
    <div class="container">
        <section class="rw-wrapper">
            </section>
        </div>
    </div>
</div>

<br>
<a href="javascript:history.go(-1)">
    <button> { { (BACK TO PREVIOUS PAGE</button>
</a>

<!--//banner-->

<div class="single">

<div class="container">
<div class="col-md-9">
    <div class="col-md-5 grid">
        <div class="flexslider">
            <ul class="slides">
                <?php echo "$goods_picture" ?>
            </ul>
        </div>
    </div>
<div class="col-md-7 single-top-in">
        <div class="span_2_of_a1 simpleCart_shelfItem">
            <?php echo "$goods_information" ?>
            <div class="clearfix"> </div>
        </div>
    </div>
</div>
<?php
$goods_information = '';
$goods_picture = '';
if (isset($_GET['gid'])) {
    $target = explode('#', $_GET['gid'])[0];
    $sql = "SELECT * FROM goods WHERE id=$target";
    if ($result = mysqli_query($con, $sql)) {
        $row = mysqli_fetch_array($result);
        $goods_information = '<h3 class="good_name">Name:'. $row['name'] .'</h3>
            <p> Description:</p>
            <p> '. $row['description'] .'</p>
            <p> '. $row['description'] .'</p>
            <p> '. $row['description'] .'</p>
            <p> '. $row['description'] .'</p>
            <p> '. $row['description'] .'</p>
            <p> '. $row['description'] .'</p>
            <div class="price_single">
            <span class="reducedfrom item_price">Present price:'. $row['present_price'] .'</span>
            </div>
            '
        $goods_picture = ' <li>The system must allow consumers</li> </ul> | PHP retrieves the cart form, filters out the qualified records |

|  |                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                        |
|--|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>what I want into the cart and browse and delete what I have in my cart.</p> | <p>to add merchandise into cart by clicking button ‘add to cart’.</p> <ul style="list-style-type: none"> <li>• The system must allow consumers to delete records in a cart by clicking on ‘delete’.</li> <li>• The system must redirect each record to detail merchandise information page.</li> <li>• The system must allow users to be redirected to cart page by clicking the button ‘cart’.</li> </ul> | <p>from the form according to the user ID obtained in the login state, and displays the records on the shopping cart page.</p> <p>Also, PHP receives post from forms to perform action of pushing cart record into orders or deleting cart record.</p> |
|--|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

PHP file: includes/cart.php and includes/cart\_data.php

Key code:

```

<?php
if(isset($_SESSION['id'])){
 include("cart_data.php");
 ?>
 <div class="col-md-10">
 <div class="check-out">
 <div class="bs-example4" data-example-id="simple-responsive-table">
 <div class="table-responsive">
 <table class="table-heading simpleCart_shelfItem">
 <tr>
 <th><?php echo "goods_list" ?></th>
 </tr>
 </table>
 </div>
 </div>
 <div class="produced">
 <div>
 <?php echo "gids" ?>
 </div>
 </div>
 </div>
 </div>
</div>

<?php
if(isset($_POST['cart_delete'])){
 if(isset($_SESSION['id'])){
 $uid = $_SESSION['id'];
 $cartid = $_POST['cartid'];
 $sql = "DELETE FROM cart WHERE cartid=$cartid";
 $check = mysqli_query($con,$sql);
 if($check){
 echo "<script>alert('Product successfully deleted from your cart!')</script>";
 echo "<script>window.location.replace('profile.php?section=cart')</script>";
 }

 }else{
 echo "<script>alert('login to delete this')</script>";
 }
}

elseif(isset($_POST['checkout']))\
 //user id
 $uid = $_SESSION['id'];

 echo "<script>console.log('user id: ' + '$uid')</script>";
 echo "<script>console.log('-----')</script>";

 //get good information, especially available dealer id and status
 if(isset($_POST['gid'])){
 $gid = $_POST['gid'];
 $cartid = $_POST['cartid'];
 for($i=0; $i<count($gid); $i++){
 echo "<script>console.log('car id: ' + '$cartid[$i]')</script>";
 $get_good = "SELECT * FROM goods WHERE gid=$gid[$i]";
 echo "<script>console.log('-----')</script>";
 //check the availability of the good to decide whether the order can be generated
 if($result_good = mysqli_query($con, $get_good)){
 $row_good= mysqli_fetch_array($result_good);
 $status = $row_good['status'];
 if($status=='available'){
 $a = $row_good['name'];
 $did = $row_good['did'];
 $gid_available = $row_good['gid'];
 echo "<script>console.log('available dealer for ' + '$a ' + 'is ' + '$did')</script>";
 $order = "INSERT INTO `orders` (`uid`, `did`, `gid`) VALUES ('$uid', '$did', '$gid_available')";
 $check = mysqli_query($con, $order);
 if($check){
 echo "<script>console.log('order generated for')</script>";
 echo "<script>console.log('good id: ' + '$gid_available')</script>";
 echo "<script>console.log('')</script>";
 }else{
 echo "<script>alert('good id:')</script>";
 echo "<script>console.log('$gid_available')</script>";
 echo "<script>console.log('is not added')</script>";
 }
 $new_cart = "DELETE FROM `cart` WHERE `cartid`=$cartid[$i]";
 $refresh_check = mysqli_query($con, $new_cart);
 }else{
 $na = $row_good['name'];
 echo "<script>alert('$na'+ ' is not available')</script>";
 }
 }
 }
 }else{
 echo "<script>console.log('goods data lost')</script>";
 }
}

echo "<script>window.location.replace('profile.php?section=orders')</script>";

```

```

 }else{
 echo "<script>console.log('goods data lost')</script>";
 }
}
echo "<script>window.location.replace('profile.php?section=orders')</script>";
}else{
 echo "<script>alert('you have nothing in cart')</script>";
}
}
?>

```

```

)else{
 echo "Please login";
}
?>

```

```

<?php
$goods_list = '<div class="mid-popular">';
$uid = $_SESSION['id'];
$gids = '<form method="post">
 <button name="checkout" class="hvr-skew-backward">Checkout</button>';
$sql = "SELECT g.*, c.name as cname, car.* FROM goods g INNER JOIN category c ON g.cid=c.id INNER JOIN cart car ON g.gid=car.gid WHERE car.uid=$uid";
if ($result = mysqli_query($con, $sql)) {
 while ($row = mysqli_fetch_array($result)) {
 $goods_list .= '<div class="col-md-3 item-grid simpleCart_shelfItem">
 <div class="mid-pop">
 <div class="pro-img">

 <div class="zoom-icon">
 <a class="picture" href="includes/detail.php?gid='.$row['gid'].'"
 rel="title" class="b-link-stripe b-animate-go thickbox"><i class="glyphicon glyphicon-search icon"></i>
 <i class="glyphicon glyphicon-menu-right icon"></i>
 </div>
 </div>
 <div class="mid-1">
 <div class="women">
 <div class="women-top"><span' . $row['cname'] .'
 <h6><a' . $row['name'] .'</h6>
 </div>
 <div class="img_item_add">
 <form method="post">
 <button name="cart_delete">DELETE</button>
 <input type="hidden" name="cartid" value= '.$row['cartid'].'></input>
 </form>
 </div>
 <div class="clearfix">
 </div>
 </div>
 </div>

```

```

 <div class="mid-2">
 <p><label>¥' . $row['original_price'] .'</label><em class="item_price">¥' . $row['present_price'] .'</p>
 </div>
 <div class="block">
 </div>
 <div class="clearfix">
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>';
 $gids .= '<input type="hidden" name="gid[]" value= '.$row['gid'].'>';
 $gids .= '<input type="hidden" name="cartid[]" value= '.$row['cartid'].'>';
}
$gids .= '</input></form>';
$goods_list .= '<div class="clearfix"></div></div>';
?>

```

## 3.8. Purchase and payment

| Function              | User story                                                   | Requirement                                                                                                                                                                                              | Implement method                                                                                                                                                                                   |
|-----------------------|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Role: Consumer</b> |                                                              |                                                                                                                                                                                                          |                                                                                                                                                                                                    |
| Purchase and payment  | As a consumer, I want to buy what I like and make a payment. | <ul style="list-style-type: none"> <li>The system must allow users to buy a product by clicking button 'Pay'.</li> <li>The system must require payment after users choose to make a purchase.</li> </ul> | <p>PHP receives the payment request.</p> <p>On the premise that the user has added the payment method, it reads the order information on the order page and updates the unpaid status to paid.</p> |

PHP file: includes/orders.php and includes/order\_data.php

Key code:

```
// pay for orders
}elseif(isset($_POST['pay'])) {
 //user id
 $uid = $_SESSION['id'];
 echo "<script>console.log('user id: ' + '$uid')</script>";
 echo "<script>console.log('-----')</script>";

 //check whether the user has payment method
 $payinfo = "SELECT paymethod FROM user WHERE id=$uid";
 if($r_payinfo = mysqli_query($con, $payinfo)){
 $row_payinfo = mysqli_fetch_array($r_payinfo)['paymethod'];
 //there is no payment method
 if($row_payinfo == 'not added'){
 echo "<script>alert('please add payment method first')</script>";
 echo "<script>window.location.replace('profile.php?section=payment','_self')</script>";
 //there is payment method
 }else{
 // update order status into paid
 if(isset($_POST['oid'])){
 $oids = $_POST['oid'];

 for($i=0; $i<count($oids); $i++){
 $sql_orders = "SELECT * FROM orders WHERE oid=$oids[$i]";
 $orders_info = mysqli_query($con, $sql_orders);
 if($row_orders=mysqli_fetch_array($orders_info)){
 $status = $row_orders['status'];
 if($status=='unpaid'){
 $set_paystatus = "UPDATE orders SET status = 'paid' WHERE oid=$oids[$i]";
 $check = mysqli_query($con,$set_paystatus);
 // update stock number
 $sql_orders = "SELECT * FROM orders WHERE oid=$oids[$i]";
 $orders_info = mysqli_query($con, $sql_orders);
 if($row_orders=mysqli_fetch_array($orders_info)){
 $gid_down = $row_orders['gid'];
 $sql_goods = "UPDATE goods SET stock = stock-1 WHERE gid=$gid_down";
 $check_goods = mysqli_query($con,$sql_goods);
 }
 }
 }
 }

 echo "<script>window.location.replace('profile.php?section=orders')</script>";
 }
 }
 }
}
```

```
}else{
 echo "<script>alert('order list is empty')</script>";
}

}

}
```

```
}else{
 echo "Please login";
}

>
```

```

<?php
$uid = $_SESSION['id'];

$name = '';
$quantity = '';
$price = '';
$status = '';
$order = '';

$order .= '<form method="post">
 <button name="pay" class="hvr-skew-backward">Pay for all</button>';
$sql = "SELECT o.*, g.gid as gid, g.name as gn, g.present_price as gp FROM orders o INNER JOIN goods g ON o.gid=g.gid WHERE o.uid=$uid";
if ($result = mysqli_query($con, $sql)) {
 $round = 0;
 while ($row = mysqli_fetch_array($result)) {
 $name .= "<form>". $row['gn']. "
</form>";
 $quantity .= "<form>". $row['num']. "
</form>";
 $price .= "<form>". $row['gp']. " * $row['num']. "
</form>";
 $status .= "<form>". $row['status']. "
</form>";
 if ($row['status'] == 'unpaid') {
 $order .= "<form method='post'><button style='font-size:12px' name='order_delete'>DELETE</button>";
 $order .= "<input type='hidden' name='oid' value= ". $row['oid'] . "></input></form>";
 } else {
 $order .= "<form method='post'><button style='font-size:12px' name='refund'>Refund</button>";
 $order .= "<input type='hidden' name='oid' value= ". $row['oid'] . "></input></form>";
 }
 $round++;

 $order .= "<input type='hidden' name='oid[]' value= ". $row['oid'] . ">";
 }
 $order .= "";
 $order .= "</input></form>";
}
?>

```

### 3.9. Refund

| Function              | User story                                                     | Requirement                                                                            | Implement method                                                                                         |
|-----------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Role: Consumer</b> |                                                                |                                                                                        |                                                                                                          |
| Refund                | As a consumer, I want to apply for refund if I change my mind. | The system must allow consumers to request refund by clicking button 'request refund'. | PHP receives post from form of each paid order record and updates the paid status into asked for refund. |

PHP file: includes/orders.php and includes/order\_data.php

Key code:



```

?>php
$uid = $_SESSION['id'];

$name = '';
$quantity = '';
$price = '';
$status = '';
$order = '';

$order .= '<form method="post">
 <button name="pay" class="hvr-skew-backward">Pay for all</button>;

$sql = "SELECT o.*, g.gid as gid, g.name as gn, g.present_price as gp FROM orders o INNER JOIN goods g ON o.gid=g.gid WHERE o.oid=$uid";
if ($result = mysqli_query($con, $sql)) {
 $count = 0;
 while ($row = mysqli_fetch_array($result)) {
 $name .= "<form>". $row['gn']. "
</form>";
 $quantity .= "<form>". $row['num']. "
</form>";
 $price .= "<form>". $row['gp'] * $row['num']. "
</form>";
 $status .= "<form>". $row['status']. "
</form>";
 if ($row['status'] == 'unpaid') {
 $order .= "<form method='post'><button style='font-size:12px' name='order_delete'>DELETE</button>";
 $order .= "<input type='hidden' name='oid' value='". $row['oid'] ."'></input></form>";
 } else {
 $order .= "<form method='post'><button style='font-size:12px' name='refund'>Refund</button>";
 $order .= "<input type='hidden' name='oid' value='". $row['oid'] ."'></input></form>";
 }
 $count++;
 $order .= "<input type='hidden' name='oid[]' value='". $row['oid'] ."'>";
 }
 $order .= "";
 $order .= "</input></form>";
}

?>

```

| Function              | User story                                     | Requirement                                                                               | Implement method                                                                                   |
|-----------------------|------------------------------------------------|-------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Role: Consumer</b> |                                                |                                                                                           |                                                                                                    |
| Add payment method    | As a consumer, I want to add a payment method. | The system must allow consumers to add payment method by filling out payment information. | PHP receives post of user input about payment method and updates the user's payment method status. |

PHP file: includes/payment.php and includes/payment\_data.php

Key code:

```
<?php include("payment_data.php");>
<div class="col-md-10">
 <div class="check-out">
 <div class="bs-example4" data-example-id="simple-responsive-table">
 <div class="table-responsive">
 <table class="table-heading simpleCart_shelfItem">
 <?php
 if(isset($_POST['edit'])){
 echo "$information_edit";
 }elseif(isset($_POST['update'])){
 $id = $_POST['id'];
 $paymethod = $_POST['paymethod'];
 echo "<script>console.log('data input received')</script>";
 $update = "UPDATE `user` SET `paymethod`='$paymethod' WHERE id=$id ";
 $check = mysqli_query($con,$update);
 if ($check) {
 echo "<script>window.location.replace('profile.php?section=payment','_self')</script>";
 echo "<script>console.log('Profile has been Updated.')</script>";
 }
 else{
 echo "<script>alert('Update failed')</script>";
 }
 }
 else{
 echo "$person_information";
 }
 >
 </table>
 </div>
 </div>
 </div>
</div>
</div>
```

```
<?php
if(isset($_SESSION['user'])){
 $uid = $_SESSION['id'];
 $sql = "SELECT * FROM user WHERE id = '$uid'";
 $result = mysqli_query($con, $sql);
 $row = mysqli_fetch_array($result);
 //account information
 $person_information = '
 <form method="post">
 <button type="submit" class="item_add hvr-skew-backward" name="edit">EDIT</button>
 </form>
 <form>
 <tr class="cart-header">
 <td>payment method</td>
 <td class="item_price">'. $row['paymethod'] .'</td>
 <td class="add-check"></td>
 </tr>
 </form>';

 //edit account information
 $information_edit = '
 <form method="post">
 <button type="submit" class="item_add hvr-skew-backward" name="update">Update</button>
 <tr class="cart-header">
 <input type="hidden" name = "id" value='.$row["id"].'></input>
 <td>payment method</td>
 <td class="item_price">'. $row['paymethod'] .'</td>
 <td class="add-check"><input name = "paymethod" value='.$row["paymethod"].'></input></td>
 </tr>
 </form>';
}
else{
 $person_information = "";
}
?>
```

## 3.11. Browse orders

| Function              | User story                                                                  | Requirement                                                                                         | Implement method                                                                                                                                                                                    |
|-----------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Role: Consumer</b> |                                                                             |                                                                                                     |                                                                                                                                                                                                     |
| Browse orders         | As a consumer, I want to browse what I have bought or requested refund for. | The system must allow consumers to access order information dashboard on personal information page. | PHP receives current user id, filters user-related order records from orders table and displays them on the orders page.<br><br>Also, PHP deletes record from orders table if receives delete post. |

PHP file: includes/orders.php and includes/order\_data.php

Key code:

```
<?php
if(isset($_SESSION['id'])) {
 include("order_data.php");
}
<div class="col-md-10">
 <div class="check-out">
 <div class="bs-example4" data-example-id="simple-responsive-table">
 <div class="table-responsive">
 <table class="table-heading simpleCart_shelfItem">
 <thead>
 <tr>
 <th>Product name</th>
 <th>Quantity</th>
 <th>Total price</th>
 <th>Status</th>
 <th>Option</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <td><?php echo "$pname" ?></td>
 <td><?php echo "$pquant" ?></td>
 <td><?php echo "$tprice" ?></td>
 <td><?php echo "$status" ?></td>
 <td><?php echo "$op" ?></td>
 </tr>
 </tbody>
 </table>
 </div>
 </div>
 <div class="produced">
 <?php echo "$oids" ?>
 </div>
 </div>
</div>
</div>
```

```

<?php
// delete order before payment
if(isset($_POST['order_delete'])){
 if(isset($_SESSION['id'])){
 $uid = $_SESSION['id'];
 $oid = $_POST['oid'];
 $sql = "DELETE FROM orders WHERE oid=$oid";
 $check = mysqli_query($con,$sql);
 if($check){
 echo "<script>alert('This order successfully deleted!')</script>";
 echo "<script>window.location.replace('profile.php?section=orders')</script>";
 }

 }else{
 echo "<script>alert('login to delete this')</script>";
 }
}

// ask for refund
}elseif(isset($_POST['refund'])){
 if(isset($_SESSION['id'])){
 $uid = $_SESSION['id'];
 $oid = $_POST['oid'];
 $sql = "UPDATE orders SET status='asked for refund' WHERE oid=$oid";
 $check = mysqli_query($con,$sql);
 if($check){
 echo "<script>alert('Successfully asked for refund!')</script>";
 echo "<script>window.location.replace('profile.php?section=orders')</script>";
 }

 }else{
 echo "<script>alert('login to delete this')</script>";
 }
}

```

```

<?php
$uid = $_SESSION['id'];

$name = '';
$quant = '';
$price = '';
$status = '';
$op = '';

$oids = '<form method="post">
 <button name="pay" class="hvr-skew-backward">Pay for all</button>';
$sql = "SELECT o.*, g.gid as gid, g.name as gn, g.present_price as gp FROM orders o INNER JOIN goods g ON o.gid=g.gid WHERE o.oid=$uid";
if ($result = mysqli_query($con, $sql)) {
 $round = 0;
 while ($row = mysqli_fetch_array($result)) {
 $name .= "<form>". $row['gn']. "
</form>";
 $quant .= "<form>". $row['num']. "
</form>";
 $price .= "<form>V". $row['gp'] * $row['num']. "
</form>";
 $status .= "<form>". $row['status']. "
</form>";
 if($row['status']=='unpaid'){
 $op .= "<form method='post'><button style='font-size:12px' name='order_delete'>DELETE</button>";
 $op .= "<input type='hidden' name='oid' value= ". $row['oid'] . "></input></form>";
 }else{
 $op .= "<form method='post'><button style='font-size:12px' name='refund'>Refund</button>";
 $op .= "<input type='hidden' name='oid' value= ". $row['oid'] . "></input></form>";
 }
 $round++;

 $oids .= "<input type='hidden' name='oid[]' value= ". $row['oid'] . ">";
 }
 $op .= "";
 $oids .= "</input></form>";
}
?>

```

### 3.12. Talk to customer services

Function	User story	Requirement	Implement method
<b>Role: Consumer</b>			
Talk to customer	As a consumer, I	Dealers contact information is displayed at the bottom of the	Dealers contact information is

services	want to talk to customer services.	home page.	displayed at the bottom of the home page.
----------	------------------------------------	------------	-------------------------------------------

PHP file: includes/templates/temp\_body\_footer.php

Key code:

```
<ul class="in in1">

contact dealer: email to XXX@XXX.com

contact admin: email to YYY@XXX.com

```

### 3.13. Browse inventory

Function	User story	Requirement	Implement method
<b>Role: Dealer</b>			
Browse inventory	As a dealer, I want to know what product I have for sale.	The system must allow dealers to browse products inventory on the index page.	PHP receives dealer id and uses this is to select dealer-related goods information including stock number from goods table.

PHP file: dealer/manage-products.php

Key code:

```

<table cellpadding="0" cellspacing="0" border="0" class="datatable-1 table table-bordered table-striped display" width="100%">
 <thead>
 <tr>
 <th>#</th>
 <th>Product Name</th>
 <th>Stock</th>
 <th>Category</th>
 <th>Status</th>
 <th>Options</th>
 </tr>
 </thead>
 <tbody>

 <?php
$query=mysqli_query($con,"select g.*, cat.name as cname from goods g inner join category cat on g.cid=cat.id where g.did=$Did ");
$cnt=1;
while($row=mysqli_fetch_array($query))
{
 ?>
 <tr>
 <td><?php echo htmlentities($cnt);?></td>
 <td><?php echo htmlentities($row['name']);?></td>
 <td><?php echo htmlentities($row['stock']);?></td>
 <td><?php echo htmlentities($row['cname']);?></td>
 <td><?php echo htmlentities($row['status']);?></td>
 <td>
 <form method="post">
 <?php
 if($row['status']=='available'){
 echo "<button name=change_ua value=".$row['gid']. ">make unavailable</button>";
 }else{
 echo "<button name=change_a value=".$row['gid']. ">make available</button>";
 }
 ?>
 </form>
 </td>
 </tr>
 <?php $cnt=$cnt+1; } ?>

```

### 3.14. Manage product status

Function	User story	Requirement	Implement method
<b>Role: Dealer</b>			
Manage product status	As a dealer, I want to edit whether a product is available to consumers.	The system must allow dealers to change the status of a product by clicking on ‘make available’ or ‘make unavailable’.	PHP receives posted good id and updates the good status as available or unavailable where good id equals to received id.

PHP file: dealer/manage-products.php

Key code:

```

<?php
//set whether the good is available to buyers
if(isset($_POST['change_ua'])){
 $gid_to_process = $_POST['change_ua'];
 $update=mysqli_query($con,"update goods set status='unavailable' where gid=$gid_to_process");
 if($update){
 echo "<script>window.location.replace('manage-products.php')</script>";
 }else{
 echo "<script>alert('update failed')</script>";
 }

 echo "processed".$gid_to_process;
}elseif(isset($_POST['change_a'])){
 $gid_to_process = $_POST['change_a'];
 $update=mysqli_query($con,"update goods set status='available' where gid=$gid_to_process");
 if($update){
 echo "<script>window.location.replace('manage-products.php')</script>";
 }else{
 echo "<script>alert('update failed')</script>";
 }

 echo "processed".$gid_to_process;
}
}
?>

```

### 3.15. Manage orders

Function	User story	Requirement	Implement method
<b>Role: Dealer</b>			
Manage orders	As a dealer, I want to deal with orders from consumers, including processing refund request.	The system must allow dealers to process refund request by clicking on process button.	PHP receives posted order id and updates the order status as request processed where order id equals to received id.

PHP file: dealer/pending-orders.php

Key code:

```

<?php
if(isset($_POST['process'])){
 $oid_to_process = $_POST['process'];
 $update=mysqli_query($con,"update orders set status='refund request processed' where oid=$oid_to_process");
 if($update){
 echo "<script>window.location.replace('pending-orders.php')</script>";
 }else{
 echo "<script>alert('update failed')</script>";
 }

 echo "processed".$oid_to_process;
}
}
?>

```

### 3.16. Browse statistics

Function	User story	Requirement	Implement method
<b>Role: Dealer</b>			
Browse statistics	As a dealer, I want to browse statistics data about my products.	The system must record each order issued by consumers and deliver them with o on the dealer page.	PHP receives posted order id and selects order record from orders table where order id equals to received id.

PHP file: dealer/pending-orders.php

Key code:

```
<table cellpadding="0" cellspacing="0" border="0" class="datatable-1" table="table-bordered table-striped display table-responsive">
<thead>
<tr>
<th>#</th>
<th>Product name</th>
<th width="50">User Contact</th>
<th>Number </th>
<th>Status </th>
<th>Options</th>
</tr>
</thead>
<tbody>
<?php
$query=mysqli_query($con,"select o.*, u.mail as umail from orders o inner join user u on o.uid=u.id where o.did=$Did");
// $query=mysqli_query($con,"select g.*, cat.name as cname from goods g inner join category cat on g.cid=cat.id where g.did=$Did");
$cnt=1;
while($row=mysqli_fetch_array($query))
{
?>
<tr>
<td><?php echo htmlentities($cnt);?></td>
<td><?php echo htmlentities($row['gid']);?></td>
<td><?php echo htmlentities($row['umail']);?></td>
<td><?php echo htmlentities($row['num']);?></td>
<td><?php echo htmlentities($row['status']);?></td>
<td><form method="post">
<?php
if($row['status']=='asked for refund'){
echo "<button name=process value='".$row['oid']."'>PROCESS</button>";
}else{
echo "Nothing to process";
}
?>
</form>
</td>
</tr>
<?php $cnt=$cnt+1;
?>
</tbody>
</table>
```



### 3.17. Browse dealer data

Function	User story	Requirement	Implement method
<b>Role: Administrator</b>			
Browse dealer data	As an administrator, I want to access data from dealers.	The system must allow the administrator to view data from dealer by clicking on hyperlinks.	PHP receives posted order id and selects order record from orders table where order id equals to received id.

PHP file: admin/manage-users.php

Key code:

```
<table cellpadding="0" cellspacing="0" border="0" class="datatable-1 table table-bordered table-striped display" width="100%">
 <thead>
 <tr>
 <th>#</th>
 <th> Location</th>
 <th> Options</th>
 </tr>
 </thead>
 <tbody>

 <?php $query=mysqli_query($con,"select * from dealer");
 $cnt=1;
 while($row=mysqli_fetch_array($query))
 {
 ?>
 <tr>
 <td><?php echo htmlentities($cnt);?></td>
 <td><?php echo htmlentities($row['location']);?></td>
 <td>
 <form method="post">
 <?php
 echo "<button name=check value='".$row['id']."'>Check order information</button>";
 ?>
 <?php
 echo "<button name=check_stock value='".$row['id']."'>Check stock information</button>";
 ?>
 </form>
 </td>
 </tr>

 <?php $cnt=$cnt+1; }

 ?>
</table>
```

```

<?php
// check order information of chosen dealer
if(isset($_POST['check'])){}
 $did_to_process = $_POST['check'];
 echo "<h3>The order information of dealer $did_to_process</h3>";
 ?>
 <table cellpadding="0" cellspacing="0" border="0" class="datatable-1 table table-bordered table-striped
display table-responsive" >
 <thead>
 <tr>
 <th>#</th>
 <th>Product name</th>
 <th width="50">User Contact</th>
 <th>Number </th>
 <th>Status </th>
 </tr>
 </thead>
 <tbody>

 <?php
$query=mysqli_query($con,"select o.*, u.mail as uemail from orders o inner join user u on o.uid=u.id
where o.did=$did_to_process");
$cnt=1;
while($row=mysqli_fetch_array($query))
{
 ?>
 <tr>
 <td><?php echo htmlentities($cnt);?></td>
 <td><?php echo htmlentities($row['gid']);?></td>
 <td><?php echo htmlentities($row['uemail']);?></td>
 <td><?php echo htmlentities($row['num']);?></td>
 <td><?php echo htmlentities($row['status']);?></td>
 </tr>

 <?php $cnt=$cnt+1;
}
 ?>
</tbody>
</table>

```

```

//check stock information of chosen dealer
}elseif(isset($_POST['check_stock'])){
 $did_to_process = $_POST['check_stock'];
 echo "<h3>The stock information of dealer $did_to_process</h3>";
 ?>
 <table cellpadding="0" cellspacing="0" border="0" class="datatable-1 table table-bordered table-striped display" width="100%">
 <thead>
 <tr>
 <th>#</th>
 <th>Product Name</th>
 <th>Stock</th>
 <th>Category</th>
 <th>Status</th>
 <th>Options</th>
 </tr>
 </thead>
 <tbody>

 <?php
$query=mysqli_query($con,"select g.*, cat.name as cname from goods g inner
join category cat on g.cid=cat.id where g.did=$did_to_process");
$cnt=1;
while($row=mysqli_fetch_array($query))
{
 ?>
 <tr>
 <td><?php echo htmlentities($cnt);?></td>
 <td><?php echo htmlentities($row['name']);?></td>
 <td><?php echo htmlentities($row['stock']);?></td>
 <td><?php echo htmlentities($row['cname']);?></td>
 <td><?php echo htmlentities($row['status']);?></td>
 <td>
 <form method="post">
 <?php
 if($row['status']=='available'){
 echo "<button name=change_ua value='".$row['gid']."'>make unavailable</button>";
 }else{
 echo "<button name=change_a value='".$row['gid']."'>make available</button>";
 }
 ?>
 </form>
 </td>
 </tr>
 <?php $cnt=$cnt+1; } ?>
</tbody>
</table>

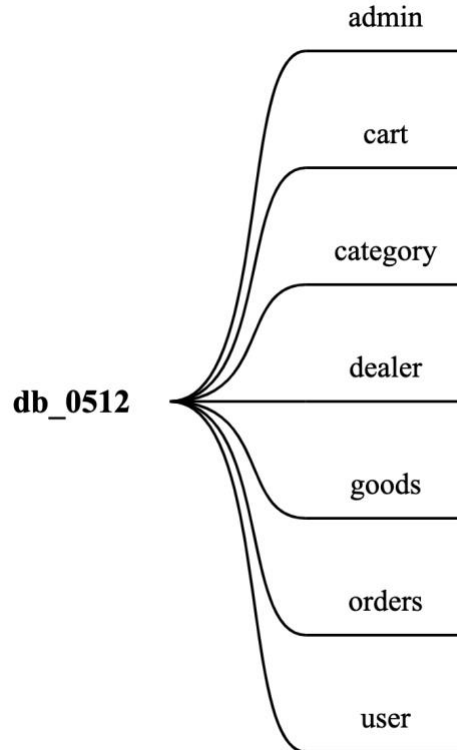
```

## 4 Data structure

This part mainly includes MySQL table structure, as well as the attributes of each item and its detailed explanation.

#### 4.1. Database structure

The following figure shows the overall database structure. The overall database is named db\_0512, which includes 7 tables.



#### 4.2. Table structure

This section shows the details of the 7 tables, including the items contained, corresponding attributes and specific explanations.

##### 1. Admin

Name	Type	Collation	Explanation
username	varchar (11)	utf8mb4_general_ci	The name of user, used for registration and login.

password	varchar (100)	utf8mb4_general_ci	The password corresponding with username, used for registration and login.
----------	---------------	--------------------	----------------------------------------------------------------------------

## 2. Cart

Name	Type	Extra	Explanation
cartid (primary)	int (11)	Auto-increment	The index value of shopping cart information, which is used to add orders later.
uid (index)	int (11)		It is a foreign key, which is used to index to user information.
gid (index)	int (11)		It is a foreign key, which is used to index to goods information.
number	int (11)		Purchase quantity, the default is 1.

## 3. Category

Name	Type	Extra	Explanation
id (primary)	int (20)	Auto-increment	This is the number of product type, which is associated with the 'cid' in the goods table, so that the product type can be displayed in the goods table.
name	varchar (20)		Used to display the product name.

#### 4. Dealer

Name	Type	Extra	Explanation
id (primary)	int (11)	Auto-increment	This is the label of dealer, which is used to associate with the 'did' in the goods and orders table so that it can display the dealer information accordingly.
location	varchar (100)		It is used to display the location information of the dealer, and it is also the login username of the dealer. For example, Shanghai.
password	varchar (100)		Dealer login password.

#### 5. Goods

Name	Type	Extra	Explanation
gid (primary)	int (20)	Auto-increment	The ID of product information, which is used to index commodity information in orders and cart table.
cid (index)	int (20)		Used to display the product category.
description	varchar (200)		Displays product description information.

name	varchar (20)		Displays product description name.
original price	int (20)		Original price of product.
present price	int (20)		Present price of product.
pic path	varchar (500)		Image path, used to display product images.
stock	int (11)		Displays the quantity of goods in stock.
did (index)	int (11)		Link dealer table to display dealer information.
status	varchar (100)		Displays the availability of the product.

## 6. Orders

Name	Type	Extra	Explanation
oid (primary)	int (11)	Auto-increment	The primary index of the order record.
uid (index)	int (11)		It associated with user table to display user information.
did (index)	int (11)		It associated with dealer table to display dealer information.
gid (index)	int (11)		It associated with goods table to display product information.
num	int (11)		Quantity of goods involved in the order. Set to 1.

status	varchar (100)		Display payment status, refund status and acceptance status.
--------	---------------	--	--------------------------------------------------------------

## 7. User

Name	Type	Extra	Explanation
id (primary)	int (20)	Auto-increment	Connect with the 'uid' in the cart and orders table to display user information in these tables.
name	varchar (20)		Store username information, which is the same as the username required for login.
phone number	varchar (20)		Store the user's contact information.
mail	varchar (100)		Store the user's mailbox information for registration and display it in the Order information on the dealer side, so that the dealer can get in touch with the user.
password	varchar (64)		Stores the user's password information for login and registration.
address	varchar (100)		Stores the user's address information for registration.

pay method	varchar (100)		Store the information of the user's payment method and use it to determine whether the payment can be made after receiving the payment request.
------------	---------------	--	-------------------------------------------------------------------------------------------------------------------------------------------------