

به نام خدا

گزارش پروژه

فاطمه بزازیان

در ابتدا وقتی سایت را باز می کنیم صفحه اصلی آن را می بینیم که این صفحه متشکل از ۳ صفحه php است. Header برای بالای صفحه، index برای بدنه صفحه اصلی و نمایش محصولات و footer برای قسمت پایین صفحه.

## فایل header.php

شکل :



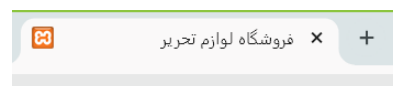
توضیح:

```
<?php  
session_start();  
?>
```

این خط یک جلسه PHP را آغاز می کند. جلسات PHP برای ذخیره سازی موقت اطلاعات کاربر در سمت سرور استفاده می شوند و برای اینکه بتوانیم با آرایه `$_SESSION` (که برای ذخیره اطلاعات کاربر به کار می رود) کار کنیم، باید دستور بالا را فراخوانی کرده و قبل آن نباید دستور HTML باشد.

```
<head>  
<meta charset="UTF-8">  
</title>فروشگاه لوازم تحریر  
<link rel="stylesheet" href="style.css">  
</head>
```

ابتدا به مرورگر اعلام می کنیم که این یک سند HTML5 است. تگ بخش `<head>` سند HTML را آغاز می کند. این بخش شامل اطلاعات متا در مورد صفحه مانند عنوان، `charset` و پیوندها به فایل های خارجی است. متا تگ به مرورگر می گوید که از رمزگذاری ی UTF-8 برای نمایش کاراکترهای صفحه استفاده کند. `<title>` عنوان صفحه را در نوار عنوان مرورگر نمایش می دهد:



و در آخر چون از یک کد CSS خارجی استفاده کردیم، این فایل را به آن ارتباط دادیم که نوع ارتباط `stylesheet` و آدرس و نام `css` خارجی در `href` مشخص شده است.

```
<body dir="rtl">  
<div class="container">  
<header>
```

```
</h1>فروشگاه لوازم تحریر</h1>
```

```
</header>
```

ابتدا در تگ `body` نوشته ها را راست چین کردیم تا از سمت راست به چپ نوشته شود. با `div` یک بخش درست می کنیم و برای استایل دهی برای آن کلاس `container` را تعریف می کنیم. از تگ `<header>` برای نمایش اطلاعات مربوط به وبسایت مانند عنوان استفاده می کنیم و از تگ `<h1>` برای نمایش عناوین اصلی استفاده می کنیم.

```
<nav>
```

```
<ul>
```

```
<li><a href="index.php">صفحه اصلی</a></li>
```

```
<li><a href="register.php">عضویت در سایت</a></li>
```

```
<?php if (isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true): ?>
```

```
<li><a href="logout.php">خروج از سایت<?php echo $_SESSION["realname"]; ?></a></li>
```

```
<?php else: ?>
```

```
<li><a href="login.php">ورود به سایت</a></li>
```

```
<?php endif; ?>
```

```
<?php if (isset($_SESSION["state_login"]) && $_SESSION["state_login"] == true && $_SESSION["user_type"] == "admin"): ?>
```

```
<li><a href="admin_products.php">مدیریت محصولات</a></li>
```

```
<li><a href="admin_manage_order.php">مدیریت سفارشات</a></li>
```

```
<?php endif; ?>
```

```
</ul>
```

```
</nav>
```

از `nav` برای ساخت منو استفاده می کنیم. `ul` یک لیست نامرتب درست می کند که حاوی آیتم های منو است. `li` آیتم ها را نمایش می دهد

صفحه اصلی    عضویت در سایت    ورود به سایت

که فردی که هنوز وارد حساب کاربری نشده تنها منوی مقابل را دارد

با کلیک بر روی هر کدام از این موارد به صفحه مورد نظر می رود مثلاً با کلیک بر روی **صفحه اصلی** دوباره به فایل `index.php` که شامل هدر و فوتر و مشخصات محصولات است برمی گردد.

در صورت کلیک بر روی **عضویت در سایت** فایل `register.php` باز می شود تا اطلاعات کاربر را بگیرد و برای او یک حساب کاربری ایجاد کند. همچنین بررسی می کند که آیا متغیر جلسه `state_login` وجود دارد و مقدار آن `true` است یا خیر. اگر این شرط درست باشد، به این معنی است که کاربر وارد سیستم شده است. بعد از ورود کاربر به حساب کاربری در قسمت منو، آیتم خروج از سایت به نمایش گذاشته می شود و نام واقعی کاربر را که در متغیر جلسه `realname` ذخیره شده است، در کنار آن چاپ می کند.

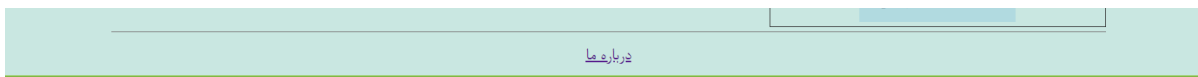
صفحه اصلی    عضویت در سایت    خروج از سایت (sima)

اگر کاربر روی منوی **ورود به سایت** کلیک کرد، فایل `login.php` برای او باز می شود که اطلاعات کاربر را گرفته و او را وارد حساب کاربری خود می کند. این شخص می تواند یا ادمین سایت باشد یا یک کاربر عادی. پس چک می کنیم متغیر جلسه `state_login` وجود داشته باشد و کاربر وارد سیستم شده باشد و اگر مقدار متغیر جلسه `user_type` برابر با `admin` باشد، آیتم های مدیریت محصولات و مدیریت سفارشات نمایش داده می شود.

صفحه اصلی    عضویت در سایت    خروج از سایت (fatemeh)    مدیریت محصولات    مدیریت سفارشات

## footer.php

شکل:



توضیح:

```
<html>
<body>
    <div class="content">
        <a href="about-us.php">در باره ما</a>
    </div>
</body>

</html>
```

در این جا یک `div` با یک بخش ایجاد می شود که متن نوشته شده در آن به صورت لینک درآمد و ما را به فایل `about-us.php` هدایت می کند.

## index.php

شکل:



توضیح:

```
<?php
include ("includes/header.php");

$link = mysqli_connect("localhost", "root", "", "shop_db");
```

```
if (mysqli_connect_errno())
    exit("خطاي با شرح زیر رخ داده است");
```

در این بخش شامل فایل header.php می‌شود. سپس با استفاده از تابع `mysqli_connect` به پایگاه داده با نام `shop_db` متصل می‌شود. در صورت عدم موفقیت در اتصال، پیام خطا نمایش داده می‌شود.

```
$query = "SELECT * FROM products";

$result = mysqli_query($link, $query);

?>

<link rel="stylesheet" type="text/css" href="styles.css">
```

در این بخش، یک پرس و جوی SQL به نام `$query` تعریف می‌شود که تمام رکوردهای موجود در جدول `products` را انتخاب می‌کند. سپس با استفاده از تابع `mysqli_query` این پرس و جو بر روی پایگاه داده (`$link`) اجرا می‌شود و نتیجه آن در متغیری به نام `$result` ذخیره می‌شود و به فایل CSS خارجی با نام `styles.css` ارتباط می‌دهد.

```
<div class="flex-container">
    <?php
    while ($row = mysqli_fetch_array($result)) {
        ?>
        <div class="product-container">
            <h4><?php echo ($row['pro_name']) ?></h4>
            " />
            <br/>
            <b>قیمت:</b> <span class="price"><?php echo ($row['pro_price']) ?> تومان</span>
            <br/>
            <b>تعداد موجودی:</b> <span class="quantity"><?php echo ($row['pro_qty']) ?></span>
            <br/>
            <b>توضیحات:</b> <span class="description"><?php echo (substr($row['pro_detail'],0,240))
?></span>
            <br/>
            <a href="order.php?id=<?php echo ($row['pro_code']) ?>" class="order-button">سفارش و خرید</a>
        </div>
    <?php
    }
    ?>
</div>
<?php
include ("includes/footer.php");
?>
```

در این بخش، یک `div` با کلاس `flex-container` ایجاد می‌شود سپس یک حلقه `while` شروع می‌شود که تا زمانی که سطر دیگری در `$result` برای خواندن وجود داشته باشد، ادامه می‌یابد. در هر تکرار از حلقه، اطلاعات مربوط به یک محصول از طریق آرایه `$row` که حاوی مقادیر ستون‌های جدول `products` است، استخراج و در عناصر `HTML` نمایش داده می‌شود.

<h4>نام محصول و <img> تصویر محصول را از پوشه `img` با نامی که در `$row['pro_image']` ذخیره شده است، نمایش می‌دهد. <span class="price"> قیمت محصول را به تومان و <span class="quantity"> تعداد موجودی محصول است.

<span class="description"> توضیحات محصول را نمایش می‌دهد <a> لینکی به صفحه `order.php` ایجاد می‌کند که `id` محصول در آن به عنوان پارامتر ارسال می‌شود و در آخر فوتر را در انتهای سایت قرار می‌دهیم.

## register.php

شکل:

توضیح:

```
<?php
include("includes/header.php");
if (isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true) {
?>
<script type="text/javascript">
location.replace("index.php");
</script>
<?php
}
```

این بخش ابتدا فایل `header.php` شامل می‌شود. سپس با استفاده از `if` شرطی بررسی می‌کند که آیا جلسه‌ای (`session`) با کلید `state_login` وجود دارد یا نه. اگر کاربر قبلاً وارد حساب کاربری خود شده باشد (جلسه `state_login` با مقدار `true` وجود داشته باشد)،

کد با استفاده از Javascript کاربر را به صفحه index.php هدایت می‌کند. اگر کاربر وارد حساب کاربری خود نشده باشد (جلسه state\_login وجود ندارد یا مقدار آن false است)، کد به سراغ فرم ثبت نام می‌رود.

```
<form name="register" action="action_register.php" method="POST">
  <table >
    <tr>
      <td style="width: 40%;"><span class="red">نام واقعی</span></td>
      <td style="width: 60%;"><input type="text" id="realname" name="realname" /></td>
    </tr>
    <tr>
      <td><span class="red">نام کاربری</span></td>
      <td><input type="text" id="username" name="username" /></td>
    </tr>
    <tr>
      <td><span class="red">کلمه عبور</span></td>
      <td><input type="password" id="password" name="password" /></td>
    </tr>
    <tr>
      <td><span class="red">تکرار کلمه عبور</span></td>
      <td><input type="password" id="repassword" name="repassword" /></td>
    </tr>
    <tr>
      <td><span class="red">پست الکترونیکی</span></td>
      <td><input type="text" id="email" name="email" /></td>
    </tr>
    <tr>
      <td><br/><br/></td>
      <td>
        <input type="submit" value="ثبت نام" />
      </td>
    </tr>
  </table>
</form>
<?php
include("includes/footer.php");
?>
```

این بخش از کد، فرم ثبت نام را با استفاده از تگ <form> به نام register برای فرم ثبت نام ایجاد می‌کند و اطلاعات فرم باید به فایل action\_register.php با روش ارسال اطلاعات POST فرستاده شود.

داخل تگ <form> یک جدول با تگ <table> برای نمایش و سازماندهی فیلدهای فرم ایجاد شده که هر سطر از جدول شامل دو ستون است. ستون اول <td> عنوان فیلد و علامت ستاره نمایش می‌دهد که نشانگر اجباری بودن آن فیلد است و ستون دوم <td> شامل المان <input> برای دریافت اطلاعات از کاربر است و یک دکمه ثبت نام داریم که اطلاعات را ارسال می‌کند و در آخر فوتر را در انتهای سایت قرار می‌دهیم.

## action\_register.php

توضیح:

```
include ("includes/header.php");
include ("includes/footer.php");
```

فایل هدر را در بالا و فوتر را در پایین صفحه مشاهده می کنیم.

```
if (isset($_POST['realname']) && !empty($_POST['realname']) &&
    isset($_POST['username']) && !empty($_POST['username']) &&
    isset($_POST['password']) && !empty($_POST['password']) &&
    isset($_POST['repassword']) && !empty($_POST['repassword']) &&
    isset($_POST['email']) && !empty($_POST['email'])) {

    $realname = $_POST['realname'];
    $username = $_POST['username'];
    $password = $_POST['password'];
    $repassword = $_POST['repassword'];
    $email = $_POST['email'];
} else
    exit("برخی از فیلدها مقدار دهی نشده است");
```

در دستور if بالا وجود و خالی نبودن فیلدهای نام واقعی، نام کاربری، رمز عبور، تکرار رمز عبور و ایمیل را بررسی می کند و مقادیر وارد شده توسط کاربر در فیلدهای مختلف را به متغیرهای مربوطه اختصاص می دهد. اگر هر یک از این فیلدها خالی باشد، اسکریپت با پیغام "برخی از فیلدها مقدار دهی نشده است" متوقف می شود.



```
if ($password != $repassword)
    exit("کلمه عبور و تکرار آن مشابه نیست");
```

این شرط بررسی می کند که آیا رمز عبور وارد شده با تکرار آن مطابقت دارد یا خیر. اگر مطابقت نداشته باشد، اسکریپت با پیغام "کلمه عبور و تکرار آن مشابه نیست" متوقف می شود.



```
if (filter_var($email, FILTER_VALIDATE_EMAIL) === false)
    exit("پست الکترونیک وارد شده صحیح نیست");
```

این خط بررسی می کند که آیا آدرس ایمیل وارد شده توسط کاربر معتبر است یا خیر. اگر آدرس ایمیل معتبر نباشد، اسکریپت با پیغام "پست الکترونیک وارد شده صحیح نیست" متوقف می شود.



```
$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno())
    exit("خطای با شرح زیر رخ داده است": mysqli_connect_error());
```

این بخش از کد به پایگاه داده MySQL با نام shop\_db با استفاده از توابع mysqli\_connect و mysqli\_connect\_errno متصل می شود. اگر اتصال به پایگاه داده با مشکلی مواجه شود، اسکریپت با پیغام خطا متوقف می شود.

```
$query = "INSERT INTO users (realname,username,password,email,type) VALUES
('$realname','$username','$password','$email','0')";
```

این خط یک پرس و جو برای افزودن اطلاعات کاربر در جدول users پایگاه داده ایجاد می کند. این پرس و جو مقادیر متغیرهای realname, username, password, email و 0 (برای نوع کاربر عادی) را به ستون های مربوطه در جدول users اضافه می کند.

```
if (mysqli_query($link, $query) === true)
{
    echo "<p class='green'><b>" . $realname .
        " گرامی عضویت شما با نام کاربری " . $username .
        " در فروشگاه با موفقیت انجام شد " . "</b></p>";
}
else
    echo "<p class='red'><b>عضویت شما در فروشگاه انجام نشد</b></p>";
mysqli_close($link);
```

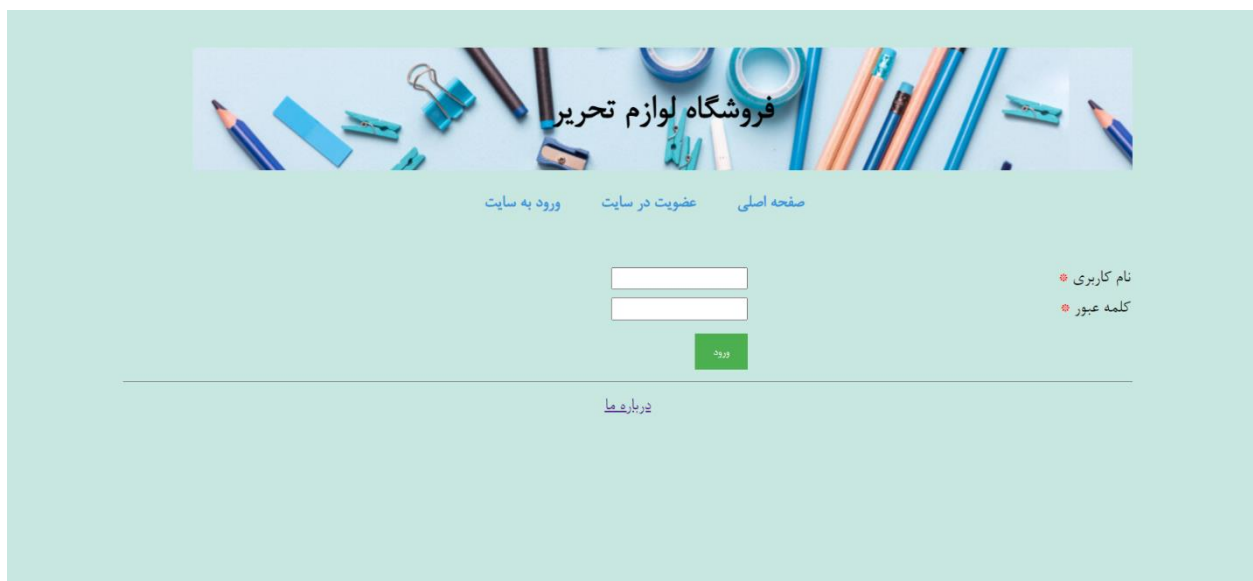
این بخش از کد، پرس و جوی SQL را با استفاده از تابع mysqli\_query اجرا می کند. اگر پرس و جو با موفقیت اجرا شود، اسکریپت با پیغام "عضویت شما با موفقیت انجام شد" به کاربر اطلاع می دهد. در غیر این صورت، اسکریپت با پیغام "عضویت شما در فروشگاه انجام نشد" به کاربر اطلاع می دهد و در آخر اتصال به پایگاه داده را با استفاده از تابع mysqli\_close می ببندد.





## Login.php

شکل:



توضیح:

```
<?php
include ("includes/header.php");
if (isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true) {
?>
<script type="text/javascript">
location.replace("index.php");
</script>
<?php
}
?>
```

در اینجا یک فایل `header.php` که شامل منو ها است در بالای صفحه به نمایش گذاشته می شود و سپس در شرط `if` بررسی می شود که آیا یک متغیر جلسه به نام `state_login` وجود دارد و مقدار متغیر جلسه `state_login` برابر با `true` است.

اگر شرط `if` برقرار باشد یعنی کاربر وارد شده و با تگ `<script>` آغاز می‌شود که از روش `location.replace` جاوا اسکریپت برای هدایت مرورگر کاربر به فایل `index.php` استفاده می‌کند و بعد از آن بلاک `if` بسته می‌شود.

```
<form name="login" action="action_login.php" method="POST">
  <table>
    <tr>
      <td><span class="red">*</span>نام کاربری</td>
      <td><input type="text" id="username" name="username" /></td>
    </tr>
    <tr>
      <td><span class="red">*</span>کلمه عبور</td>
      <td><input type="password" id="password" name="password" /></td>
    </tr>
    <tr>
      <td><br /><br /></td>
      <td><input type="submit" value="ورود" /></td>
    </tr>
  </table>
</form>
```

این بخش از کد یک فرم ورود به سیستم را نمایش می‌دهد، اما تنها زمانی که کاربر قبلاً وارد حساب کاربری خود نشده باشد (شرط `if` برقرار نباشد) قابل مشاهده است. فرم با تگ `<form>` شروع می‌شود که نام آن `login` و صفحه‌ای که فرم هنگام ارسال داده‌ها به آن هدایت می‌شود `action_login.php` و روشی که داده‌های فرم با آن ارسال می‌شوند `POST` است.

داخل فرم، یک جدول برای نمایش فیلدهای ورود به سیستم وجود دارد. دو فیلد با تگ‌های `<input>` برای نام کاربری و کلمه عبور ساخته می‌شوند. یک دکمه ایجاد می‌کنیم برای ارسال فرم.

```
<?php
include ("includes/footer.php");
?>
```

این خط یک فایل `PHP` دیگر به نام `footer.php` را در این اسکریپت شامل می‌کند.

## action\_login.php

شکل:



توضیح:

```
include ("includes/header.php");
include ("includes/footer.php");
```

در این صفحه نیز هدر در بالا و فوتر در پایین صفحه قرار می گیرد.

```
if ((isset($_POST['username']) && !empty($_POST['username']) &&
isset($_POST['password']) &&
!empty($_POST['password']))) {

    $username = $_POST['username'];
    $password = $_POST['password'];
} else
    exit("برخی از فیلد ها مقدار دهی نشده است");
```

در شرط if وجود و عدم خالی بودن فیلدهای نام کاربری و رمز عبور را بررسی می کند و مقادیر وارد شده را به متغیرهای مربوطه اختصاص می دهد. اگر هر یک از این فیلدها خالی باشد، اسکریپت با پیغام "برخی از فیلد ها مقدار دهی نشده است" متوقف می شود.



```
$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno())
    exit(mysqli_connect_error() . " : خطای با شرح زیر رخ داده است");
```

این بخش از کد به پایگاه داده MySQL با نام "shop\_db" با استفاده از توابع `mysqli_connect` و `mysqli_connect_errno` متصل می شود و اگر اتصال به پایگاه داده با مشکلی مواجه شود، اسکریپت با پیغام خطا متوقف می شود.

```
$query = "SELECT * FROM users WHERE username='$username' AND
password='$password'";
$result = mysqli_query($link, $query);
```

این خط یک پرس و جوی SQL را برای انتخاب اطلاعات کاربر از جدول `users` در پایگاه داده ایجاد می کند. این پرس و جو رکوردهایی را از جدول `users` انتخاب می کند که در آن نام کاربری با `username` وارد شده توسط کاربر و رمز عبور با `password` وارد شده توسط کاربر مطابقت داشته باشد و در ادامه پرس و جوی SQL را با استفاده از تابع `mysqli_query` اجرا می کند و نتیجه پرس و جو در متغیر `result` ذخیره می شود.

```
$row = mysqli_fetch_array($result);
```

این خط از تابع `mysqli_fetch_array` برای بازیابی اولین رکورد از مجموعه نتایج ذخیره شده در متغیر `result` استفاده می کند. این رکورد در یک آرایه به نام `row` ذخیره می شود.

```
if ($row) {
    $_SESSION["state_login"] = true;
    $_SESSION["realname"] = $row['realname'];
    $_SESSION["username"] = $row['username'];
}
```

```

if ($row["type"] == 0)
    $_SESSION["user_type"] = "public";
elseif ($row["type"] == 1) {
    $_SESSION["user_type"] = "admin";
?>
<script type="text/javascript">
location.replace("admin_products.php");
</script>

```

این بخش از کد بررسی می کند که آیا متغیر row شامل داده ای هست یا خیر. اگر row شامل داده باشد، به این معنی است که کاربر با موفقیت وارد شده است. اگر ورود به سیستم موفقیت آمیز باشد، این بخش از کد اطلاعات کاربر را در متغیرهای جلسه ذخیره می کند. `$_SESSION["user_type"]` نوع کاربری (مانند عمومی یا مدیر) را از جدول `users` ذخیره می کند. بر اساس نوع کاربری ذخیره شده در جلسه، کاربر به صفحه مناسب هدایت می شود. اگر از نوع `admin` بود، کاربر با استفاده از `JavaScript` به صفحه `admin_products.php` هدایت می شود. در غیر این صورت، کاربر در همان صفحه با پیغام خوش آمدید روبرو می شود.

```

<?php
}
echo ("<p class='green'><b> کاربر {$row['realname']} به فروشگاه لوازم تحریر خوش آمدید</b></p>");
} else
echo ("<p class='red'><b>نام کاربری یا کلمه عبور یافت نشد</b></p>");
mysqli_close($link); include ("includes/footer.php");
include ("includes/footer.php");
?>

```

این بخش از کد بر اساس موفقیت یا عدم موفقیت ورود به سیستم، یک پیام به کاربر نمایش می دهد و در آخر اتصال به پایگاه داده را با استفاده از تابع `mysqli_close` می بندد و فوتر را نمایش می دهد.

## logout.php

شکل:



توضیح:

بعد از کلیک بر روی این قسمت کاربر بلافاصله از حساب کاربری خود خارج می شود.

```
session_start();
```

در اینجا یک جلسه PHP را آغاز می شود. جلسات PHP برای ذخیره سازی موقت اطلاعات کاربر در سمت سرور استفاده می شوند.

```
session_unset();
```

این خط همه متغیرهای جلسه را که با جلسه فعلی مرتبط هستند، پاکسازی می کند. این کار هرگونه اطلاعات خاص کاربر ذخیره شده در جلسه را حذف می کند.

```
session_destroy();
```

این خط جلسه PHP فعلی را کاملاً از بین می‌برد. این امر به طور مؤثر به جلسه پایان می‌دهد و هرگونه داده جلسه مرتبط با کاربر را حذف می‌کند.

```
<script type="text/javascript">
location.replace("index.php");
</script>
```

این کد یک بلوک کد JavaScript را آغاز می‌کند و از روش `location.replace` جاوا اسکریپت برای هدایت مرورگر کاربر به فایل `index.php` استفاده می‌کند.

## order.php

توضیح :

```
<?php
include("includes/header.php");

$link = mysqli_connect("localhost", "root", "", "shop_db");

if (mysqli_connect_errno()) {
    exit("خطای زیر رخ داده است: " . mysqli_connect_error());
}
```

در بخش های قبلی توضیح دادم.

```
$pro_code = 0;
if (isset($_GET['id'])) {
    $pro_code = $_GET['id'];
}
```

در این بخش، ابتدا متغیری به نام `pro_code` برای ذخیره کد محصول تعریف و مقدار آن به ۰ تنظیم می شود سپس، با استفاده از `isset($_GET['id'])` بررسی می شود که آیا پارامتر `id` در `$_GET` وجود دارد یا نه. اگر پارامتر `id` وجود داشته باشد، مقدار آن به `pro_code` با استفاده از `$_GET['id']` انتقال داده می شود.

```
if (!isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true)) {
?>

<div class="container">
    <br />
    <span>!</span> برای خرید پستی محصول انتخاب شده باید وارد سایت شوید
    <br/>
    در صورتی که عضو فروشگاه هستید برای تکمیل خرید خود روی
    <a href='login.php' style='text-decoration: none;' ><span class='blue'><b>ورود به
سایت</b></span></a>
    کلیک کنید
    <br/>
    در صورتی که عضو فروشگاه هستید برای تکمیل خرید خود روی
```

```

        <a href='register.php' style='text-decoration: none;' ><span class='blue'><b>عضویت در
سایت</b></span></a>
        کلیک کنید
        <br />
    </div>
<?php
    exit();
}

```

در این بخش، بررسی می شود که آیا کلید state\_login در جلسه (\$\_SESSION) وجود دارد یا نه و همچنین بررسی می شود که مقدار کلید state\_login در جلسه true باشد. اگر هر دوی این شرایط false باشند (یعنی کاربر وارد نشده باشد)، کد داخل if اجرا می شود. یک div با کلاس container برای قالب بندی محتوا ایجاد می شود. یک پیام با مضمون! برای خرید پستی محصول انتخاب شده باید وارد سایت شوید! نمایش داده می شود. دو لینک با متن ورود به سایت و عضویت در سایت به ترتیب به صفحات login.php و register.php ایجاد می شود. در نهایت، اسکریپت با exit() متوقف می شود.



```

$query = "SELECT * FROM products WHERE pro_code='".$pro_code'";
$result = mysqli_query($link, $query);
$row = mysqli_fetch_array($result);
?>

```

در این بخش، یک عبارت SQL SELECT در متغیر \$query ذخیره می شود. این عبارت تمام ستون های (\*) از جدول products را انتخاب می کند. به شرطی که فقط سطرهایی را با pro\_code مطابق با \$pro\_code برمی گرداند. \$query با استفاده از mysqli\_query روی اتصال به پایگاه داده اجرا می شود. نتیجه اجرای عبارت SQL در متغیر \$result ذخیره می شود و از mysqli\_fetch\_array برای استخراج اولین سطر از مجموعه نتایج ذخیره شده در \$result استفاده می شود. آرایه استخراج شده در متغیر \$row ذخیره می شود.

```

<div class="container">
    <form name="order" action="action_order.php" method="POST">
        <table class="form-table">
            <?php
            if ($row) {
                <tr>
                    <td><span class="red">*</span></td>

```

```

        <td><input type="text" id="pro_code" name="pro_code" value="<?php echo ($pro_code);
?>" class="input-style" readonly /></td>
    </tr>
    <tr>
        <td><span class="red">نام کالا</span></td>
        <td><input type="text" id="pro_name" name="pro_name" value="<?php echo
($row['pro_name']); ?>" class="input-style" readonly /></td>
    </tr>
    <tr>
        <td><span class="red">تعداد درخواستی</span></td>
        <td><input type="text" id="pro_qty" name="pro_qty" onchange="calc_price();" /></td>
    </tr>
    <tr>
        <td><span class="red">قیمت واحد کالا</span></td>
        <td><input type="text" id="pro_price" name="pro_price" value="<?php echo
($row['pro_price']); ?>" class="input-style" readonly />تومان</td>
    </tr>
    <tr>
        <td><span class="red">مبلغ قابل پرداخت</span></td>
        <td><input type="text" id="total_price" name="total_price" value="0" class="input-
style" readonly />تومان</td>
    </tr>
    <?php
    $user_query = "SELECT * FROM users WHERE username='{$_SESSION['username']}'";
    $user_result = mysqli_query($link, $user_query);
    $user_row = mysqli_fetch_array($user_result);
    ?>
    <tr>
        <td><span class="red">نام خریدار</span></td>
        <td><input type="text" id="realname" name="realname" value="<?php echo
($user_row['realname']); ?>" class="input-style" readonly /></td>
    </tr>
    <tr>
        <td><span class="red">پست الکترونیکی</span></td>
        <td><input type="text" id="email" name="email" value="<?php echo ($user_row['email']);
?>" class="input-style" readonly /></td>
    </tr>
    <tr>
        <td><span class="red">شماره همراه</span></td>
        <td><input type="text" id="mobile" name="mobile" value="09" /></td>
    </tr>
    <tr>
        <td><span class="red">آدرس پستی</span></td>
        <td><textarea id="address" name="address" cols="30" rows="3"></textarea></td>
    </tr>

```

```

        <tr>
            <td><br /><br /></td>
            <td><input type="button" value="خرید محصول" onclick="check_input();" /></td>
        </tr>
    <?php
    }
    ?>
</table>
</form>
</div>

```

در این بخش از کد، فرمی را با استفاده از تگ <form> به نام order ایجاد می‌کنیم و اطلاعات فرم باید به فایل action\_order.php با روش ارسال اطلاعات POST فرستاده شود. با استفاده از if (\$row)، بررسی می‌کنیم که آیا اطلاعات محصول (\$row) از کد قبلی استخراج شده است یا نه. اگر \$row مقداری داشته باشد (یعنی محصول با pro\_code معتبر پیدا شده باشد)، کد داخل if اجرا می‌شود. داخل if، چندین سطر جدول برای ایجاد فیلدهای فرم با اطلاعات محصول وجود دارد. کد کالا (pro\_code) و نام کالا (pro\_name) از اطلاعات محصول (\$row) استخراج شده و در فیلدهای فقط خواندنی نمایش داده می‌شوند. فیلد pro\_qty (تعداد درخواستی) تابع calc\_price را برای محاسبه قیمت کل تنظیم شده است. قیمت واحد (pro\_price) از اطلاعات محصول استخراج شده و با واحد تومان نمایش داده می‌شود.

فیلد مبلغ قابل پرداخت مقدار اولیه ۰ تومان دارد و به صورت فقط خواندنی است تا بعداً با محاسبه مقدار دهی شود.

سپس یک پرس و جو SQL برای بازیابی اطلاعات کاربر از یک جدول users ایجاد می‌شود که تمام ستون‌ها را از جدول users انتخاب می‌کند و نتایجی را که فقط شامل رکورد کاربری که در آن ستون username با مقدار ذخیره شده در متغیر جلسه \$\_SESSION['username'] مطابقت دارد، فیلتر می‌کند. سپس پرس و جوی SQL ذخیره شده در متغیر \$user\_query را با استفاده از تابع mysqli\_query اجرا می‌کند سپس سعی می‌کند اولین ردیف داده را از \$user\_result با استفاده از تابع mysqli\_fetch\_array بازیابی کند.

سایر اطلاعات کاربر نیز گرفته می‌شود. نام و ایمیل کاربر را که از پایگاه داده بازیابی شده است، نمایش داده می‌شود و یک عنصر ورودی با شناسه mobile با مقدار اولیه ۰۹ ایجاد می‌شود و یک عنصر textarea با شناسه address و نام address ایجاد می‌شود که به کاربران اجازه می‌دهد آدرس کامل خود را وارد کنند دارای ۳۰ ستون و ۳ ردیف است.

یک عنصر دکمه با مقدار "خرید محصول" ایجاد می‌شود. رویداد کلیک آن یک تابع JavaScript به نام check\_input را فعال می‌کند.



<input type="text" value="۱۳"/>	کد کالا *
<input type="text" value="گیره کالبد"/>	نام کالا *
<input type="text"/>	تعداد درخواستی *
<input type="text" value="۳۵۰۰۰"/>	قیمت واحد کالا *
<input type="text" value="تومان"/>	
<input type="text" value="۰"/>	مبلغ قابل پرداخت *
<input type="text" value="تومان"/>	
<input type="text" value="sara moradi"/>	نام خریدار *
<input type="text" value="sara@irib.ir"/>	پست الکترونیکی *
<input type="text" value="۰۹"/>	شماره همراه *
<input type="text"/>	آدرس پستی *

```
<script type="text/javascript">
function calc_price() {
    var pro_qty = <?php echo ($row['pro_qty']); ?>;
    var price = document.getElementById('pro_price').value;
    var count = document.getElementById('pro_qty').value;
    var total_price;
    if (count > pro_qty) {
        alert('!!تعداد موجودی انبار کمتر از درخواست شما است');
        document.getElementById('pro_qty').value = 0;
        count = 0;
    }
    if (count == 0 || count == '') {
        total_price = 0;
    } else {
        total_price = count * price;
    }
    document.getElementById('total_price').value = total_price;
}
}
```

وظیفه تابع `function calc_price()` محاسبه قیمت کل (`total_price`) بر اساس تعداد درخواستی (`count`) و قیمت واحد (`price`) را دارد. متغیر `pro_qty` از مقدار استخراج شده از `<?php echo ($row['pro_qty']); ?>` برای حداکثر تعداد موجودی استفاده می کند. مقادیر `price` و `count` از فیلدهای `input` مربوطه در فرم با استفاده از `document.getElementById` گرفته می شوند.

یک شرط `if` بررسی می کند که آیا `count` بیشتر از `pro_qty` است یا نه. اگر `true` باشد، به کاربر هشدار می دهد و `count` و `total_price` را به ۰ تنظیم می کند. در غیر این صورت، `total_price` با ضرب `count` در `price` محاسبه می شود. در نهایت،

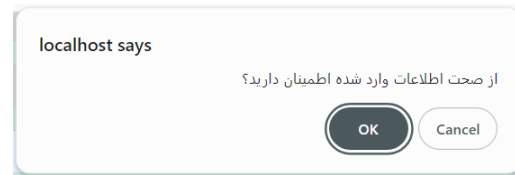
total\_price به روز شده در فیلد input مربوطه (total\_price) با استفاده از document.getElementById تنظیم می شود. خروجی:



```
function check_input() {
    var r = confirm("از صحت اطلاعات وارد شده اطمینان دارید؟");
    if (r == true) {
        var validation = true;
        var count = document.getElementById('pro_qty').value;
        var mobile = document.getElementById('mobile').value;
        var address = document.getElementById('address').value;
        if (count == 0 || count == '') {
            validation = false;
        }
        if (mobile.length < 11) {
            validation = false;
        }
        if (validation) {
            document.order.submit();
        } else {
            alert('برخی از ورودی های فرم سفارش محصول به درستی پر نشده اند');
        }
    }
}
</script>
<?php
include ("includes/footer.php");
?>
```

وظیفه تابع function check\_input() بررسی ورودی های کاربر و ارسال فرم در صورت معتبر بودن است. از confirm برای نمایش یک پنجره تأیید به کاربر استفاده می شود. اگر کاربر تأیید را انتخاب کند (r == true)، کد داخل if اجرا می شود validation برای نشان دادن این که آیا تمام ورودی ها معتبر هستند یا نه، ابتدا true در نظر گرفته می شود. مقادیر count، mobile و address از فیلدهای input مربوطه در فرم با استفاده از document.getElementById گرفته می شوند.

if اول بررسی می کند که آیا count (تعداد درخواستی) خالی یا ۰ است (اعتبار ندارد). if دوم بررسی می کند که آیا mobile (شماره همراه) کمتر از ۱۱ رقم دارد (نامعتبر). اگر هر یک از شرایط false باشد، validation به false تغییر می یابد. if سوم، اگر validation true باشد، فرم با استفاده از document.order.submit() ارسال می شود. در غیر این صورت، یک پیغام هشدار به کاربر نمایش داده می شود.



## action\_order.php

توضیح:

```
<?php
include ("includes/header.php");
if (!(isset($_SESSION["state_login"]) && $_SESSION["state_login"] == true)) {
?>
<script type="text/javascript">
location.replace("index.php");
</script>
<?php
```

این کد را قبلا توضیح دادم.

```
$pro_code = $_POST['pro_code'];
$pro_name = $_POST['pro_name'];
$pro_qty = $_POST['pro_qty'];
$pro_price = $_POST['pro_price'];
$total_price = $_POST['total_price'];
$realname = $_POST['realname'];
$email = $_POST['email'];
$mobile = $_POST['mobile'];
$address = $_POST['address'];
$username = $_SESSION['username'];
$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno())
    exit("خطای با شرح زیر رخ داده است:" . mysqli_connect_error());
```

این بخش کد، اطلاعات ارسال شده از فرم را در متغیرهای PHP ذخیره می کند. این متغیرها به ترتیب کد محصول، نام محصول، تعداد محصول، قیمت واحد، قیمت کل، نام، ایمیل، شماره تلفن، آدرس و نام کاربری کاربر را دریافت می کند و به پایگاه داده متصل می شود و بررسی می شود که آیا هنگام برقراری اتصال به پایگاه داده خطایی رخ داده است یا نه اگر تابع `mysqli_connect_errno` مقداری غیر صفر برگرداند، یعنی خطایی رخ داده است در این صورت کد با نمایش یک پیام خروج که شامل توضیح خطا `mysqli_connect_error()` است، متوقف می شود.

```
$query = "INSERT INTO orders
(id,
username,
orderdate,
pro_code,
```

```

pro_qty,
pro_price,
mobile,
address
) VALUES
('0',
 '$username',
 '".date('y-m-d')." ',
 '$pro_code',
 '$pro_qty',
 '$pro_price',
 '$mobile',
 '$address'
);

```

در این بخش دستور INSERT را داریم که برای درج رکورد جدید در جدول orders است که لیستی از فیلدهای جدول orders را که می خواهیم مقادیر جدید را در آنها وارد کنیم، مشخص می کنیم سپس بعد از کلمه کلیدی VALUES که نشان دهنده شروع لیست مقادیر برای درج رکورد در فیلدهای جدول است، مقادیر جدیدی را که می خواهیم در فیلدهای مربوطه درج کنیم، لیست می کنیم. مقدار فیلد username از متغیر \$username که قبلا تعریف شده، درج می شود و برای فیلد orderdate از تابع date با فرمت y-m-d استفاده می شود و سایر فیلدها مقادیر خود را از متغیرهای PHP مربوطه که قبلا تعریف شده اند، دریافت می کنند.

```

if (mysqli_query($link, $query) === true) {
    echo ("<p><br><b>سفارش شما با موفقیت ثبت شد</b></p>");
    echo ("<p><br><b>$realname کاربر گرامی</b></p>");
    echo ("<p><br><b>تومان را $pro_price با قیمت پایه $pro_qty به تعداد $pro_code با کد $pro_name محصول</b></p>");
    echo ("<p><br><b>سفارش داده اید</b></p>");
    echo ("<p><br><b>$total_price مبلغ قابل پرداخت</b></p>");
    echo ("<p><b>پس از بررسی سفارش و تایید آن با شما تماس گرفته خواهد شد</b></p>");
    echo ("<p><b>محصول خریداری شده از طریق پست به آدرس درج شده ارسال خواهد شد</b></p>");
    echo ("<p><b>در هنگام تحویل محصول آن را بررسی و از سالم بودن آن اطمینان حاصل کنید سپس مبلغ کالا به مامور پست تحویل</b></p>");
    $query = "UPDATE products SET pro_qty=pro_qty-$pro_qty WHERE pro_code='$pro_code'";
    mysqli_query($link, $query);
} else
    echo ("<p><b>خطا در ثبت سفارش</b></p>");

mysqli_close($link);
include ("includes/footer.php");
?>

```

در این بخش از تابع mysqli\_query برای اجرای عبارت \$query (که قبلا برای INSERT و درج اطلاعات سفارش تعریف شده بود) استفاده می کند که اگر با موفقیت اجرا شود، کد داخل بلوک if اجرا می شود که به کاربر اطلاع می دهد که سفارش با موفقیت ثبت شده است و نام کاربری را نمایش می دهد و سپس جزئیات سفارش شامل نام محصول، کد محصول، تعداد، و قیمت واحد و در آخر اطلاعاتی در مورد نحوه بررسی سفارش، ارسال محصول و تحویل آن به کاربر ارائه می دهند :



و در ادامه کد موجودی کالا به روزرسانی می شود به این صورت که موجودی فعلی (pro\_qty) از تعداد سفارش داده شده (\$pro\_qty) کم می شود و موجودی فقط برای محصولی با کد \$pro\_code که در سفارش ذکر شده، به روز می شود.

در صورتی که عبارت SQL INSERT با خطا مواجه شود، کد داخل بلوک else اجرا می شود و یک پیام مبنی بر خطا در ثبت سفارش به کاربر نمایش داده می شود و در آخر اتصال به پایگاه داده را با استفاده از تابع mysqli\_close می بندیم و فوتر را در انتهای صفحه قرار می دهیم.

## admin\_products.php

توضیح:

```
<?php
include("includes/header.php");
if (!(isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true && $_SESSION["user_type"] == "admin")) {
    ?>
    <script type="text/javascript">
        location.replace("index.php");
    </script>
    <?php
        exit();
}
$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno()) {
    exit("خطای با شرح زیر رخ داده است: " . mysqli_connect_error());
}
```

کدهای بالا قبلاً توضیح داده شده اند.

```
$url = $pro_code = $pro_name = $pro_qty = $pro_price = $pro_image = $pro_detail = "";
$btn_caption = "افزودن کالا";
```

در ابتدای این کد چند متغیر را تعریف و مقدار اولیه آنها را خالی یا ("") قرار می دهیم و مقدار پیش فرض "افزودن کالا" را برای متغیر \$btn\_caption تنظیم می کنیم که نشان می دهد فرم به طور پیش فرض برای افزودن محصول جدید استفاده می شود.

```
if (isset($_GET['action']) && $_GET['action'] == 'EDIT') {
    $id = $_GET['id'];
    $query = "SELECT * FROM products WHERE pro_code='$id'";
    $result = mysqli_query($link, $query);
    if ($row = mysqli_fetch_array($result)) {
        $pro_code = $row['pro_code'];
        $pro_name = $row['pro_name'];
        $pro_qty = $row['pro_qty'];
        $pro_price = $row['pro_price'];
        $pro_image = $row['pro_image'];
        $pro_detail = $row['pro_detail'];
        $url = "?id=$pro_code&action=EDIT";
        $btn_caption = "ویرایش کالا";
    }
}
```

در شرط if بررسی می کنیم که آیا کلید action در آرایه \$\_GET وجود دارد یا نه و همینطور چک می شود که آیا مقدار کلید action برابر با EDIT است یا نه. اگر هر دو شرط درست باشند، به این معنی است که کاربر در حال ویرایش یک محصول موجود است و کد داخل شرط if اجرا می شود. در این شرط پارامتر id (که همان کد محصولی است که قرار است ویرایش شود) را از رشته \$\_GET در متغیر \$id ذخیره می کند و یک عبارت SQL برای انتخاب تمام فیلدها (\*) از جدول products در جاهایی که کد محصول با مقدار \$id مطابقت دارد، ایجاد می کند و با استفاده از اتصال به پایگاه داده آن را اجرا می کند و نتیجه را در متغیر \$result ذخیره می کند.

همچنین در شرط if دوم، تابع mysqli\_fetch\_array اولین ردیف از نتیجه \$result را بازیابی می کند و آن را به صورت یک آرایه در متغیر \$row قرار می دهد که اگر \$row مقداری داشته باشد (یعنی اطلاعات محصول با کد \$id پیدا شده باشد)، اطلاعات مربوط به محصول را از ردیف بازیابی شده (\$row) استخراج کرده و در متغیرهای مربوطه ذخیره می کند و یک رشته URL برای فرم ویرایش محصول

ایجاد می کند که این URL شامل پارامترهای id (با مقدار کد محصول) و action (با مقدار EDIT) است. همچنین مقدار متغیر \$btn\_caption را به "ویرایش کالا" تغییر می دهد، که نشان می دهد فرم در حال حاضر برای ویرایش یک محصول موجود استفاده می شود.

```
<form name="add_product" action="action_admin_products.php<?php if (!empty($url)) echo($url); ?>" method="POST"
enctype="multipart/form-data">
    <table>
        <tr>
            <td><span class="red">*</span></td>
            <td><input type="text" id="pro_code" name="pro_code" value="<?php echo htmlspecialchars($pro_code); ?>"
/></td>
        </tr>
        <tr>
            <td><span class="red">نام کالا</span></td>
            <td><input type="text" id="pro_name" name="pro_name" value="<?php echo htmlspecialchars($pro_name); ?>"
/></td>
        </tr>
        <tr>
            <td><span class="red">موجودی کالا</span></td>
            <td><input type="text" id="pro_qty" name="pro_qty" value="<?php echo htmlspecialchars($pro_qty); ?>" /></td>
        </tr>
        <tr>
            <td><span class="red">قیمت کالا</span></td>
            <td><input type="text" id="pro_price" name="pro_price" value="<?php echo htmlspecialchars($pro_price); ?>" />
تومان</td>
        </tr>
        <tr>
            <td><span class="red">آپلود تصویر کالا</span></td>
            <td>
                <input type="file" id="pro_image" name="pro_image" size="30" />
                <?php if (!empty($pro_image)) echo ("<img src='img/' . htmlspecialchars($pro_image) . "' alt='Product
Image' />"); ?>
            </td>
        </tr>
        <tr>
            <td><span class="red">توضیحات تکمیلی کالا</span></td>
            <td><textarea id="pro_detail" name="pro_detail" cols="45" rows="10" wrap="virtual"><?php echo
htmlspecialchars($pro_detail); ?></textarea></td>
        </tr>
        <tr>
            <td><br /><br /></td>
            <td><input type="submit" value="<?php echo htmlspecialchars($btn_caption); ?>" /></td>
        </tr>
    </table>
</form>
```

این بخش از کد، با استفاده از تگ `<form>` به نام `add_product` فرمی برای افزودن کالا به فروشگاه ایجاد می‌کند و اطلاعات فرم باید به آدرس نوشته شده در بخش `action` با روش ارسال اطلاعات `POST` فرستاده شود که داده ها را به صورت پنهان ارسال می‌کند. در ادامه بررسی می‌شود که آیا متغیر `$url` مقداری دارد یا نه. اگر مقدار داشته باشد، آن مقدار را به آدرس `action` ارسال می‌کند. این کار باعث می‌شود فرم بداند که در حال ویرایش محصول است و از کدام محصول باید اطلاعات را ویرایش کند و همچنین ویژگی `enctype="multipart/form-data"` برای فرم هایی که شامل آپلود فایل هستند، ضروری است

داخل تگ `<form>` یک جدول با تگ `<table>` برای نمایش و سازماندهی فیلدهای فرم ایجاد شده که هر سطر حاوی دو ستون است. ستون اول نام فیلد و یک ستاره قرمز برای نشان دادن فیلدهای اجباری دارد و ستون دوم المان ورودی فرم را نمایش می‌دهد.

کد محصول، نام کالا، موجودی، قیمت و توضیحات تکمیلی به ترتیب با تگ های `input` و `textarea` نمایش داده می‌شوند.

"`<?php echo htmlspecialchars($pro_code); ?>`" `value=` مقدار فعلی فیلد کد محصول از متغیر `$pro_code` را در تگ `value` قرار می‌دهد. این کار باعث می‌شود که در صورت ویرایش محصول، کد قبلا وارد شده نمایش داده شود.

به همین ترتیب، مقادیر سایر فیلدها (`$pro_name`, `$pro_qty`, `$pro_price`, `$pro_detail`) نیز از متغیرهای PHP مربوطه در تگ های `value` قرار می‌گیرند.

فیلد آپلود تصویر با تگ `input` از نوع `file` نمایش داده می‌شود و بررسی می‌شود که آیا متغیر `$pro_image` مقداری دارد یا نه (یعنی تصویری آپلود شده است). اگر مقداری داشته باشد، یک تگ `img` برای نمایش تصویر ایجاد می‌کند و `alt` متن جایگزین برای تصویر در صورت عدم نمایش تصویر است و در آخر یک دکمه `submit` برای ارسال اطلاعات فرم داریم که مقدار دکمه (`value`) از متغیر `$btn_caption` گرفته می‌شود که بسته به اینکه فرم برای افزودن محصول جدید یا ویرایش محصول موجود استفاده می‌شود، تغییر می‌کند.

```
<?php
$query = "SELECT * FROM products";
$result = mysqli_query($link, $query);
?>
<table class="products">
    <tr>
```



```

<th>کد کالا</th>
<th>نام کالا</th>
<th>موجودی کالا</th>
<th>قیمت کالا</th>
<th>تصویر کالا</th>
<th>ایز از مدیریتی</th>
</tr>
<?php
while ($row = mysqli_fetch_array($result)) {
?>
<tr>
<td><?php echo htmlspecialchars($row['pro_code']); ?></td>
<td><?php echo htmlspecialchars($row['pro_name']); ?></td>
<td><?php echo htmlspecialchars($row['pro_qty']); ?></td>
<td><?php echo htmlspecialchars($row['pro_price']); ?>&nbsp;تومان</td>
<td></td>
<td>
<b><a href="action_admin_products.php?id=<?php echo htmlspecialchars($row['pro_code']);
?>&action=DELETE">حذف</a></b>
&nbsp;|&nbsp;
<b><a href="admin_products.php?id=<?php echo htmlspecialchars($row['pro_code']);
?>&action=EDIT">ویرایش</a></b>
</td>
</tr>
</tr>
<?php
}
?>
</table>
<?php
include("includes/footer.php");
?>
</body>
</html>

```

این بخش کد، یک عبارت SQL برای انتخاب تمام فیلدها (\*) از جدول products را در متغیر \$query ذخیره می کند. سپس با استفاده از تابع mysqli\_query و اتصال به پایگاه داده (\$link)، این عبارت را اجرا می کند و نتیجه را در متغیر \$result ذخیره می کند.

یک جدول ایجاد می کنیم که سربرگ جدول شامل ۶ عنوان است. سپس یک حلقه ایجاد می کنیم که در هر تکرار حلقه، تابع mysqli\_fetch\_array یک ردیف از \$result را بازیابی می کند و آن را به صورت یک آرایه در متغیر \$row قرار می دهد.

تا زمانی که ردیفی برای بازیابی باقی مانده باشد، حلقه while اجرا می شود و برای هر محصول، یک سطر جدید در جدول ایجاد می شود و برای هر ستون جدول مقدار مربوط به فیلد محصول از آرایه \$row بازیابی می شود و با استفاده از echo نمایش داده می شود. مسیر تصویر در پوشه img قرار دارد و نام تصویر از فیلد pro\_image در \$row می گیرد.

ستون آخر حاوی لینک هایی برای حذف و ویرایش محصول است:

لینک حذف به صفحه `action_admin_products.php` با پارامترهای کد محصول (`id`) و `action` با مقدار `DELETE` هدایت می شود.

لینک ویرایش به صفحه `admin_products.php` با پارامترهای کد محصول (`id`) و `action` با مقدار `EDIT` هدایت می شود و بعد از آن حلقه به پایان می رسد و فوتر را در انتهای صفحه قرار می دهیم.

کد کالا	نام کالا	موجودی کالا	قیمت کالا	تصویر کالا	ابزار مدیریتی
۱	ماژیک وایت برد	۳۵	۱۵۰۰۰ تومان		حذف   ویرایش
۲	غلط گیر	۶۶	۳۰۰۰۰ تومان		حذف   ویرایش
۳	پاک کن	۲۳۰	۸۰۰۰۰ تومان		حذف   ویرایش
۴	ماژیک هایلایتر	۴۸۰	۵۵۰۰۰۰ تومان		حذف   ویرایش

## action\_admin\_products.php

توضیح:

```
<?php
include ("includes/header.php");
if (!(isset($_SESSION["state_login"]) && $_SESSION["state_login"] === true &&
$_SESSION["user_type"] ==
    "admin")) {
?>
<script type="text/javascript">
location.replace("index.php");
</script>
<?php
}
```

این بخش قبلاً توضیح داده شده است.

```
if (!(isset($_GET['action']) && $_GET['action']=='DELETE')){
    if (isset($_POST['pro_code']) && !empty($_POST['pro_code']) && isset($_POST['pro_name']) &&
        !empty($_POST['pro_name']) && isset($_POST['pro_qty']) && !empty($_POST['pro_qty']) &&
        isset($_POST['pro_price']) && !empty($_POST['pro_price']) && isset($_POST['pro_detail']) &&
        !empty($_POST['pro_detail'])) {
```

```

$pro_code = $_POST['pro_code'];
$pro_name = $_POST['pro_name'];
$pro_qty = $_POST['pro_qty'];
$pro_price = $_POST['pro_price'];
$pro_image = basename($_FILES["pro_image"]["name"]);
$pro_detail = $_POST['pro_detail'];
} else
    exit("برخی از فیلدها مقدار دهی نشده است");
}

```

این بخش کد، ابتدا بررسی می کند که آیا درخواست حذف محصول با استفاده از پارامترهای action و DELETE در URL (\$\_GET) وجود دارد یا خیر

این بخش بررسی می کند که آیا کلید action در آرایه \$\_GET وجود دارد یا نه و همچنین بررسی می کند که آیا مقدار کلید action برابر با 'DELETE' است یا نه. اگر درخواست حذف محصول وجود داشته باشد شرط if اجرا می شود در غیر این صورت، فرم اضافه یا ویرایش محصول را بررسی می کنیم.

در شرط if بررسی می کند که آیا کلید pro\_code در آرایه \$\_POST وجود دارد یا نه و همچنین بررسی می کند که آیا مقدار کلید pro\_code در \$\_POST خالی است یا نه. به همین ترتیب، بقیه فیلدهای فرم نیز بررسی می شوند. اگر تمام فیلدها ارسال شده باشند و مقادیر خالی نباشند، کد داخل این if اجرا می شود که می گوید مقادیر ارسال شده توسط فرم در متغیرهای مربوطه ذخیره شوند و نام فایل آپلود شده از طریق \$\_FILES با استفاده از تابع basename استخراج شود و اگر تمام فیلدها ارسال نشده باشند یا مقادیر خالی باشند، کد داخل else اجرا می شود. در این کد، یک پیام خطا مبنی بر "برخی از فیلدها مقدار دهی نشده است" با استفاده از exit نمایش داده می شود.

```

$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno())
    exit("خطای با شرح زیر رخ داده است");

```

این بخش را قبل توضیح دادم.

```

if (isset($_GET['action'])) {
    $id = $_GET['id'];
    switch ($_GET['action']) {
        case 'EDIT':
            $query = "UPDATE products SET
                pro_code='$pro_code',
                pro_name='$pro_name',
                pro_qty='$pro_qty',
                pro_price='$pro_price',
                pro_detail='$pro_detail'
            WHERE pro_code='$id'";
            if (mysqli_query($link, $query) === true)
                echo "<p class='red'><b>محصول انتخاب شده با موفقیت ویرایش شد</b></p>";
            else
                echo "<p class='red'><b>خطا در ویرایش محصول</b></p>";
            break;
    }
}

```

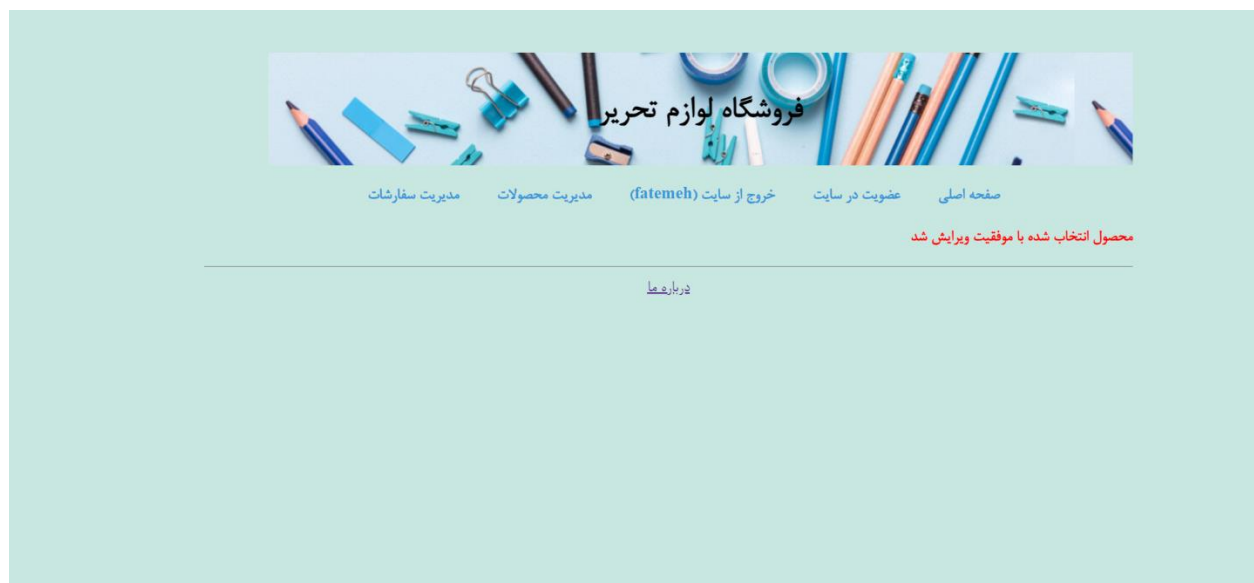
```

case 'DELETE':
    $query = "SELECT pro_image FROM products
    WHERE pro_code='$id'";
    $result=mysqli_query($link, $query);
    $row = mysqli_fetch_array($result);
    $pro_image=$row['pro_image'];
    $query = "DELETE FROM products
    WHERE pro_code='$id'";
    if (mysqli_query($link, $query) === true){
        echo ("<p class='green'><b>محصول انتخاب شده با موفقیت حذف شد</b></p>");
        unlink("img/".$pro_image);
    }
    else
        echo ("<p class='red'><b>خطا در حذف محصول</b></p>");
    break;
}
mysqli_close($link);
include ("includes/footer.php");
exit();
}

```

این بخش کد، بررسی می کند که آیا پارامتر **action** که نشان دهنده درخواست ویرایش یا حذف است، در **\$\_GET** وجود دارد یا نه و از یک سوئیچ برای انجام عملیات ویرایش یا حذف بر اساس مقدار **action** استفاده می کند به این صورت که اگر عملیات ویرایش انتخاب شده بود، کد موجود در **case 'EDIT'** اجرا می شود. در این کد **\$id** از پارامتر **id** در **\$\_GET['id']** برای شناسایی محصول مورد نظر برای ویرایش بازیابی می شود و یک عبارت **SQL** برای به روز رسانی اطلاعات محصول در جدول **products** با مقادیر جدید از **\$\_POST** و با استفاده از **\$id** برای **WHERE** ایجاد می شود. اگر عبارت **SQL** با موفقیت اجرا شود، پیامی مبنی بر موفقیت ویرایش نمایش داده می شود و در غیر این صورت، پیام خطا نمایش داده می شود.

در نهایت، اتصال به پایگاه داده با **mysqli\_close** بسته می شود و فوتر صفحه را قرار می دهیم و اسکریپت با **exit** متوقف می شود.



اگر عملیات حذف انتخاب شده بود، کد موجود در 'DELETE' case اجرا می شود. در این کد \$id از پارامتر id در \$\_GET['id'] برای شناسایی محصول مورد نظر برای حذف بازیابی می شود. ابتدا، تصویر محصول بازیابی می شود سپس دو عبارت SQL برای حذف محصول از جدول products و همچنین حذف فایل تصویر آپلود شده با استفاده از unlink اجرا می شوند. اگر عملیات حذف با موفقیت انجام شود، پیامی مبنی بر موفقیت حذف نمایش داده می شود. در غیر این صورت، پیام خطا نمایش داده می شود.

در نهایت، اتصال به پایگاه داده با mysqli\_close بسته می شود و فوتر صفحه را قرار می دهیم و اسکریپت با exit متوقف می شود.



```
$target_dir = "img/";
$target_file = $target_dir.basename($_FILES["pro_image"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);
$check = getimagesize($_FILES["pro_image"]["tmp_name"]);
```

در این بخش، دو متغیر برای آپلود تصویر تعریف می شود :

متغیر \$target\_dir مسیر دایرکتوری برای ذخیره فایل آپلود شده را دارد و متغیر \$target\_file نام کامل فایل آپلود شده با ترکیب \$target\_dir و نام پایه فایل. متغیری به نام \$uploadOk برای نشان دادن وضعیت آپلود ۱ به معنی آپلود موفق ایجاد می شود. سپس، پسوند فایل (\$imageFileType) با استفاده از pathinfo استخراج می شود. بعد از آن، از تابع getimagesize برای بررسی نوع MIME و ابعاد تصویر استفاده می شود.

```
if ($check != false) {
    echo "<br /> " . $check["mime"] . " - پرونده انتخابی یک تصویر از نوع " . $check["mime"] . " است";
    $uploadOk = 1;
}
else {
    echo "<br /> پرونده انتخاب شده یک تصویر نیست ";
    $uploadOk = 0;
}
```

اگر \$check مقدار false برگرداند، نشان دهنده عدم وجود تصویر یا تصویر نامعتبر است. در غیر این صورت، نوع MIME تصویر در \$check["mime"] نمایش داده می شود و \$uploadOk روی ۱ باقی می ماند.

```
if (file_exists($target_file)) {
    echo "<br /> پرونده ای با همین نام در سرور وجود دارد";
}
```

```
$uploadOk = 0;
}
```

در این بخش، با استفاده از `file_exists` بررسی می شود که آیا فایلی با نام `$target_file` (مسیر و نام کامل فایل) در هاست وجود دارد یا خیر. اگر فایل هم نامی وجود داشته باشد، پیغامی مبنی بر وجود فایل با همین نام نمایش داده می شود و `$uploadOk` روی ۰ تنظیم می شود.



```
if ($_FILES["pro_image"]["size"] > 500000) {
    echo " <br />اندازه پرونده انتخابی بیشتر از ۵۰۰ کیلوبایت است ";
    $uploadOk = 0;
}
```

در این بخش، حجم فایل آپلود شده بررسی می شود. اگر حجم فایل از ۵۰۰,۰۰۰ بایت بیشتر باشد، پیغامی مبنی بر بزرگ بودن حجم فایل به نمایش در می آید و `$uploadOk` روی ۰ تنظیم می شود.

```
if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg") {
    echo " <br />برای پرونده مجاز هستند JPG, JPEG, PNG فقط پسوند های ";
    $uploadOk = 0;
}
```

در این بخش، فرمت تصویر (`$imageFileType`) با فرمت های مجاز (`jpg, jpeg, png`) مقایسه می شود. اگر فرمت تصویر با هیچ یک از فرمت های مجاز مطابقت نداشته باشد، پیغامی مبنی بر نامعتبر بودن فرمت و `$uploadOk` به ۰ نمایش داده می شود.

```
if ($uploadOk == 0) {
    echo " <br />پرونده انتخاب شده به سرویس دهنده میزبان ارسال نشد ";
} else {
    if (move_uploaded_file($_FILES["pro_image"]["tmp_name"], $target_file)) {
        echo " . basename($_FILES["pro_image"]["name"]) . " <br />با موفقیت به سرویس دهنده میزبان انتقال یافت ";
    } else {
        echo " <br />خطا در ارسال پرونده به سرویس دهنده میزبان رخ داده است ";
    }
}
```

در این بخش، اگر `$uploadOk` برابر با ۱ باشد (یعنی تمام بررسی ها با موفقیت انجام شده است)، از تابع `move_uploaded_file` برای آپلود فایل از محل موقت (`$_FILES["pro_image"]["tmp_name"]`) به مسیر نهایی (`$target_file`) استفاده می شود. اگر آپلود موفقیت آمیز باشد، پیغامی مبنی بر آپلود موفق و نام فایل نمایش داده می شود. در غیر این صورت، پیغام خطای آپلود نمایش داده می شود.

```
if ($uploadOk == 1) {
    $query = "INSERT INTO products
    (pro_code,
    pro_name,
    pro_qty,
    pro_price,
    pro_image,
    pro_detail) VALUES
    ('$pro_code',
    '$pro_name',
    '$pro_qty',
    '$pro_price',
    '$pro_image',
    '$pro_detail')";
    if (mysqli_query($link, $query) === true)
        echo ("<p class='green'><b>کالا با موفقیت به انبار اضافه شد</b></p>");
    else
        echo ("<p class='red'><b>خطا در ثبت مشخصات کالا در انبار</b></p>");
} else
    echo ("<p class='red'><b>خطا در ثبت مشخصات کالا در انبار</b></p>");
mysqli_close($link);

include ("includes/footer.php");
?>
```

در این بخش، اگر آپلود (`$uploadOk`) با موفقیت انجام شده باشد، اطلاعات محصول به همراه نام فایل آپلود شده (`$pro_image`) در پایگاه داده ذخیره می شود. یک عبارت `SQL INSERT` برای درج اطلاعات محصول در جدول `products` با استفاده از متغیرهای موجود برای نام تصویر ایجاد می شود. اگر عبارت `SQL` با موفقیت اجرا شود، پیغامی مبنی بر ذخیره موفقیت آمیز محصول نمایش داده می شود. در غیر این صورت، پیغام خطای مربوط به پایگاه داده نمایش داده می شود.





در نهایت، اتصال به پایگاه داده با `mysqli_close` بسته می شود و فوتر صفحه را قرار می دهیم.

## admin\_manage\_order.php

شکل:

کد سفارش	نام خریدار	نام محصول	تاریخ سفارش	تعداد سفارش	قیمت کالا	قیمت نهایی	شماره تماس	آدرس
۱	sara moradi	غلط گیر	۲۰۲۴-۰۶-۰۹	۶۸	۲۰۰۰۰ تومان	۱۳۶۰۰۰۰ تومان	۰۹۱۶۳۴۱۰۲۷۲	تهران
۲	sima	مداد تراش	۲۰۲۴-۰۶-۱۴	۲	۱۰۰۰۰ تومان	۲۰۰۰۰ تومان	۰۹۰۳۲۲۲۲۲۲۲	شیراز
۳	sima	مداد تراش	۲۰۲۴-۰۶-۱۵	۳	۱۰۰۰۰ تومان	۳۰۰۰۰ تومان	۰۹۱۳۱۱۱۲۲۱۱	مازندران
۴	sima	جامدادی	۲۰۲۴-۰۶-۱۵	۵	۳۰۰۰۰ تومان	۱۵۰۰۰۰ تومان	۰۹۱۱۲۲۲۳۳۴۴	اراک
۵	sima	غلط گیر	۲۰۲۴-۰۶-۱۵	۱	۲۰۰۰۰ تومان	۲۰۰۰۰ تومان	۰۹۱۳۱۱۱۲۲۱۱	زاهدان

توضیح:

```
<?php
include("includes/header.php");

if (!(isset($_SESSION["state_login"]) && $_SESSION["state_login"] == true &&
$_SESSION["user_type"] == "admin")) {
    ?>
    <script type="text/javascript">
```



```

        location.replace("index.php");
    </script>
    <?php
    exit;
}
$link = mysqli_connect("localhost", "root", "", "shop_db");
if (mysqli_connect_errno()) {
    exit("خطایی در اتصال به پایگاه داده رخ داده است");
}
$query = "SELECT * FROM orders";
$result = mysqli_query($link, $query);
?>
<link rel="stylesheet" type="text/css" href="styles.css">

```

کدهای بالا را در قسمت های قبلی توضیح دادم.

```

<table class="table-order" >
    <tr>
        <th>کد سفارش</th>
        <th>نام خریدار</th>
        <th>نام محصول</th>
        <th>تاریخ سفارش</th>
        <th>تعداد سفارش</th>
        <th>قیمت کالا</th>
        <th>قیمت نهایی</th>
        <th>شماره تماس</th>
        <th>آدرس</th>
    </tr>

```

کد بالا با تگ <table> یک جدول ایجاد می کند که در سطر اول آن ۸ سرستون وجود دارد که عنوان هر ستون را نشان می دهد و متن نوشته شده در آن به صورت bold است.

```

<?php
while ($row = mysqli_fetch_array($result)) {
    $query_user = "SELECT * FROM users WHERE username='{$row['username']}'";
    $result_user = mysqli_query($link, $query_user);
    $row_user = mysqli_fetch_array($result_user);

    $query_product = "SELECT * FROM products WHERE pro_code='{$row['pro_code']}'";
    $result_product = mysqli_query($link, $query_product);
    $row_product = mysqli_fetch_array($result_product);
}
?>

```

در کد بالا یک حلقه شروع می شود که این حلقه تا زمانی که اطلاعات بیشتری برای پردازش از result وجود داشته باشد، اجرا می شود. در هر بار تکرار حلقه، یک ردیف از داده ها از result با استفاده از تابع mysqli\_fetch\_array به متغیر \$row اختصاص داده می شود.

بخش بعدی یک عبارت SQL ایجاد می کند برای انتخاب تمام فیلدها از جدول users در جایی که نام کاربری با نام کاربری موجود در ردیف فعلی مطابقت داشته باشد و در ادامه SQL تعریف شده در \$query\_user را با استفاده از اتصال دیتابیس اجرا می کند و نتیجه را در متغیر \$result\_user ذخیره می شود.

بخش بعدی اولین ردیف از نتیجه \$result\_user را بازیابی کرده و آن را به متغیر \$row\_user اختصاص می دهد. این ردیف حاوی اطلاعات کاربر مرتبط با سفارش است.

در ۳ خط بعد فرآیند مشابهی برای استخراج اطلاعات محصول انجام می شود با جایگزینی users با products و username با pro\_code.

```
<tr>
    <td><?php echo $row['id']; ?></td>
    <td><?php echo $row_user['realname']; ?></td>
    <td><?php echo $row_product['pro_name']; ?></td>
    <td><?php echo $row['orderdate']; ?></td>
    <td><?php echo $row['pro_qty']; ?></td>
    <td><?php echo $row['pro_price']; ?>&nbsp;تومان</td>
    <td><?php echo $row['pro_qty'] * $row['pro_price']; ?>&nbsp;تومان</td>
    <td><?php echo $row['mobile']; ?></td>
    <td><?php echo $row['address']; ?></td>
</tr>
<tr class="footer-row">
    <td colspan="9"></td>
</tr>
<?php
```

این بخش اطلاعات موجود در \$row، \$row\_user و \$row\_product را برای نمایش در یک ردیف جدول قالب بندی می کند سپس کد مقدار فیلد id از ردیف فعلی را در سلول جدول نمایش می دهد به همین ترتیب، مقادیر فیلدهای realname از \$row\_user، pro\_name از \$row\_product، orderdate، pro\_qty، pro\_price، mobile و address از \$row نمایش داده می شوند و مقدار pro\_qty در pro\_price ضرب می شوند و حاصل را به عنوان قیمت کل سفارش به همراه واحد تومان نمایش می دهد و یک ردیف خالی در انتهای هر گروه از ردیف های داده برای جدا کردن بخش های مختلف در جدول اضافه می شود و در آخر حلقه while را می بندیم.

about-us.php

شکل:



توضیح:

```
<section id="about">
  <h2>درباره ما</h2>
  <p>. این سایت برای راحتی شما در خرید طراحی شده است. امیدواریم تجربه خوبی در استفاده از آن داشته باشید.</p>
</section>
```

در اینجا یک بخش با شناسه about ایجاد می‌شود که شامل یک عنوان <h2> با متن "درباره ما" و یک پاراگراف همراه با متن آن است.

```
section id="contact">
  <h2>تماس با ما</h2>
  <p>. ایمیل بزنید <a href="mailto:example@example.com">example@example.com</a> جهت ارتباط با ما می‌توانید به آدرس ایمیل </p>
</section>
```

بعد از آن، یک بخش دیگر با شناسه contact ایجاد می‌شود که این بخش هم شامل یک عنوان <h2> با متن "تماس با ما" و یک پاراگراف همراه با متن آن است.