**توضیحات و گزارش پروژه(سایت خریید کیف و کفش)**

**راحله اقبالی 4001231003**

**فایل config.php**

این قطعه کد PHP به اتصال به پایگاه داده MySQL با استفاده از توابع mysqli\_connect می‌پردازد.

1. $serverip="localhost";

این خط کد آدرس IP یا نام میزبان سرور MySQL را تعیین می‌کند. در اینجا، مقدار localhost برای استفاده از سرور MySQL محلی (Localhost) استفاده شده است.

2. $dbname="kif&kafsh";

این خط کد نام پایگاه داده‌ای که قصد اتصال به آن را دارید را تعیین می‌کند. در اینجا، نام پایگاه داده kif&kafsh است.

3. $username="root";

این خط کد نام کاربری برای ورود به پایگاه داده MySQL را تعیین می‌کند. در اینجا، نام کاربری root استفاده شده است که ممکن است برای محیط توسعه‌ای باشد و در محیط تولیدی توصیه نمی‌شود.

4. $password="";

این خط کد رمز عبور برای ورود به پایگاه داده MySQL را تعیین می‌کند. در اینجا، رمز عبور خالی است که مناسب برای محیط توسعه‌ای است و در محیط تولیدی توصیه نمی‌شود.

5. $conn = mysqli\_connect($serverip,$username,$password,$dbname);

این خط کد اتصال به پایگاه داده MySQL را برقرار می‌کند با استفاده از تابع mysqli\_connect و پارامترهای آدرس IP یا نام میزبان سرور، نام کاربری، رمز عبور و نام پایگاه داده.

6. سه خط بعدی (mysqli\_query($conn,"SET NAMES 'utf8'"), mysqli\_query($conn,"SET CHARACTER SET 'utf8'"), mysqli\_query($conn,"SET character\_set\_connection = 'utf8'"))

این سه خط کد تنظیمات مختلفی را برای برقراری ارتباط با پایگاه داده MySQL انجام می‌دهند، به طور کلی تنظیمات زبان و کاراکتر برای پشتیبانی از UTF-8 را تعیین می‌کنند.

با استفاده از این قطعه کد، یک اتصال به پایگاه داده MySQL برقرار شده و تنظیمات لازم برای پشتیبانی از UTF-8 اعمال شده است.

**فایل header.php**

این قطعه کد PHP برای احراز هویت کاربران و مدیریت سبد خرید در یک وبسایت استفاده می‌شود.

1. if (!isset($\_SESSION)){ session\_start(); }

این خط کد یک نشست (session) را شروع می‌کند. نشست‌ها برای ذخیره اطلاعات کاربر در طول زمان استفاده می‌شوند و برای احراز هویت و موارد دیگر در وبسایت‌ها استفاده می‌شوند.

2. require\_once("config.php");

این خط کد فایل config.php را وارد می‌کند که شامل تنظیمات اتصال به پایگاه داده MySQL است.

3. if(isset($\_POST['email'])){ ... }

این قسمت کد بررسی می‌کند که آیا فرم ورود به سیستم توسط کاربر ارسال شده است یا خیر. اگر ارسال شده باشد، ایمیل و رمز عبور وارد شده توسط کاربر را دریافت می‌کند و یک کوئری SQL برای جستجو در جدول user اجرا می‌کند تا کاربر با این اطلاعات یافت شود.

4. $query = "SELECT \* FROM user WHERE email='$email' AND password='$password'";

در این خط کد، یک کوئری SQL برای جستجو در جدول user بر اساس ایمیل و رمز عبور وارد شده توسط کاربر تعریف می‌شود.

5. $result = mysqli\_query($conn,$query);

این خط کد کوئری SQL را اجرا می‌کند و نتیجه را در متغیر $result ذخیره می‌کند.

6. if(mysqli\_num\_rows($result) > 0){ ... } else { ... }

در این بخش، بررسی می‌شود که آیا کاربر با این اطلاعات یافت شده یا خیر. اگر یافت شده باشد، اطلاعات کاربر (مانند ایمیل و نام) در نشست ذخیره می‌شود و کاربر به صفحه اصلی هدایت می‌شود. در غیر اینصورت، کاربر به صفحه ورود به سیستم با پیام خطا منتقل می‌شود.

7. if(isset($\_GET['id\_cart'])){ ... }

در این بخش، بررسی می‌شود که آیا کاربر یک محصول به سبد خرید خود اضافه کرده است یا خیر. اگر محصول اضافه شده باشد، تعداد آن در سبد خرید ذخیره می‌شود.

با استفاده از این قطعه کد، احراز هویت کاربران و مدیریت سبد خرید در وبسایت انجام می‌شود.

و در باقی فایل ها مثل ایندکس و ابوت هدر را فراخوانی کردیم.

**فایل login.php**

این قطعه کد PHP برای احراز هویت کاربران و انتقال آن‌ها به صفحه اصلی یا صفحه ورود با پیام خطا استفاده می‌شود.

1. session\_start();

این خط کد یک نشست (session) را شروع می‌کند تا اطلاعات کاربر در طول زمان ذخیره شود.

2. require\_once("config.php");

فایل config.php که شامل تنظیمات اتصال به پایگاه داده MySQL است را وارد می‌کند.

3. if(isset($\_POST['email'])){ ... }

این قسمت کد بررسی می‌کند که آیا فرم ورود به سیستم توسط کاربر ارسال شده است یا خیر. اگر ارسال شده باشد، ایمیل و رمز عبور وارد شده توسط کاربر را دریافت می‌کند و یک کوئری SQL برای جستجو در جدول user اجرا می‌کند.

4. $row = mysqli\_fetch\_assoc($result);

این خط کد نتیجه کوئری SQL را به صورت یک آرایه از داده‌ها استخراج می‌کند.

5. if(mysqli\_num\_rows($result) > 0){ ... } else { ... }

این بخش بررسی می‌کند که آیا کاربر با اطلاعات وارد شده یافت شده یا خیر. اگر یافت شده باشد، اطلاعات کاربر (مانند ایمیل و نام) در نشست ذخیره می‌شود و کاربر به صفحه اصلی هدایت می‌شود. در غیر اینصورت، کاربر به صفحه ورود با پیام خطا منتقل می‌شود.

6. $emilGet = ''; if(isset($\_GET['email'])){ $emilGet = $\_GET['email']; }

در این بخش، بررسی می‌شود که آیا پارامتر email در URL وجود دارد یا خیر. اگر وجود داشته باشد، مقدار آن در متغیر $emilGet ذخیره می‌شود.

با استفاده از این قطعه کد، احراز هویت کاربران و انتقال آن‌ها به صفحه اصلی یا صفحه ورود با پیام خطا انجام می‌شود.

**فایل logout.php**

1. session\_start();: در این خط، یک نشست جدید برای کاربر شروع می‌شود. اگر نشست قبلاً شروع شده باشد، از آن استفاده می‌شود.

2. session\_destroy();: این دستور باعث حذف تمامی متغیرهای مربوط به نشست فعلی کاربر می‌شود و نشست را تماماً پاک می‌کند.

3. header("location: index.php");: این دستور به مرورگر می‌گوید که به آدرس index.php هدایت شود. به این ترتیب، پس از حذف نشست، کاربر به صفحه اصلی وبسایت فروشگاه هدایت می‌شود.

به طور کلی، این کد PHP در فایل logout.php وظیفه حذف نشست کاربر (خروج از حساب کاربری) و هدایت کاربر به صفحه اصلی وبسایت را بر عهده دارد.

**فایل cart.php**

این قطعه کد PHP برای مدیریت سبد خرید کاربران استفاده می‌شود. زمانی که کاربر یک محصول را از سبد خرید حذف می‌کند یا سبد خرید را خالی می‌کند، این قطعه کد وظیفه حذف محصول یا خالی کردن سبد خرید را بر عهده دارد.

1. session\_start();

این خط کد یک نشست (session) را شروع می‌کند تا اطلاعات کاربر در طول زمان ذخیره شود.

2. require\_once("config.php");

فایل config.php که شامل تنظیمات اتصال به پایگاه داده MySQL است را وارد می‌کند.

3. if(isset($\_GET['deleteId'])) { ... }

این بخش کد بررسی می‌کند که آیا پارامتر deleteId در URL وجود دارد یا خیر. اگر وجود داشته باشد، محصول مورد نظر از سبد خرید حذف می‌شود و تعداد کل محصولات در سبد خرید کاهش می‌یابد.

4. unset($\_SESSION['products'][$\_GET['deleteId']]);

با استفاده از تابع unset، محصول مورد نظر از آرایه $\_SESSION['products'] حذف می‌شود.

5. $\_SESSION['count\_cart'] -=1;

تعداد کل محصولات در سبد خرید کاهش می‌یابد.

6. if(isset($\_GET['emptyCart'])) { ... }

در این بخش، بررسی می‌شود که آیا پارامتر emptyCart در URL وجود دارد یا خیر. اگر وجود داشته باشد، تمامی محصولات از سبد خرید حذف می‌شوند و تعداد کل محصولات در سبد خرید صفر می‌شود.

7. unset($\_SESSION['products']);

با استفاده از تابع unset، تمامی محصولات از آرایه $\_SESSION['products'] حذف می‌شوند.

8. $\_SESSION['count\_cart'] = 0;

تعداد کل محصولات در سبد خرید صفر می‌شود.

با استفاده از این قطعه کد، حذف محصولات از سبد خرید و خالی کردن سبد خرید در نشست کاربران مدیریت می‌شود.

**فایل single.php**

1. require\_once("config.php");

این خط کد فایل config.php را وارد می‌کند که شامل تنظیمات اتصال به پایگاه داده MySQL است.

2. if(isset($\_GET['p\_id'])){ ... }

این بخش کد بررسی می‌کند که آیا پارامتر p\_id در URL وجود دارد یا خیر. اگر وجود داشته باشد، کوئری برای دریافت اطلاعات محصول مربوط به p\_id اجرا می‌شود.

3. $query = "SELECT \* FROM products WHERE product\_id=".$\_GET['p\_id'];

در این بخش، یک کوئری SQL برای دریافت تمامی اطلاعات محصول با شناسه p\_id ایجاد می‌شود.

4. $result = mysqli\_query($conn,$query);

کوئری SQL با استفاده از تابع mysqli\_query اجرا می‌شود و نتیجه آن در متغیر $result ذخیره می‌شود.

5. $row = mysqli\_fetch\_assoc($result);

با استفاده از تابع mysqli\_fetch\_assoc، اطلاعات محصول با شناسه p\_id از نتیجه کوئری دریافت شده و در آرایه $row ذخیره می‌شود.

با استفاده از این قطعه کد، اطلاعات محصول مورد نظر از پایگاه داده با استفاده از شناسه p\_id در URL دریافت می‌شود.

**فایل register.php**

این قطعه کد PHP برای ثبت نام کاربران در سیستم استفاده می‌شود. زیرا این کد از پایگاه داده MySQL استفاده می‌کند و اطلاعات را به صورت امن درون آن ذخیره می‌کند. حالتی که کاربر وجود داشته باشد یا نداشته باشد نیز بررسی می‌شود.

1. session\_start();

این دستور به منظور شروع یک session استفاده می‌شود تا بتوان اطلاعات کاربر را در سراسر صفحات نگهداری کرد.

2. require\_once ('config.php');

این خط کد فایل config.php را وارد می‌کند که شامل تنظیمات اتصال به پایگاه داده MySQL است.

3. if(isset($\_POST['email'])){ ... }

این بخش کد بررسی می‌کند که آیا اطلاعات فرم ثبت نام (ایمیل، رمز عبور، نام، نام خانوادگی و آدرس) ارسال شده‌اند یا خیر.

4. $selectEmail = "SELECT \* FROM user WHERE email='$email'";

در این بخش، یک کوئری SQL برای جستجوی کاربر با ایمیل مشخص شده ایجاد می‌شود.

5. $exist = mysqli\_query($conn,$selectEmail);

کوئری SQL برای جستجو ایمیل در جدول کاربران اجرا می‌شود و نتیجه آن در $exist ذخیره می‌شود.

6. if(mysqli\_num\_rows($exist) > 0){ ... }

اگر تعداد ردیف‌های بازگشتی بیشتر از صفر باشد، یعنی کاربر با این ایمیل قبلاً ثبت نام کرده است، بنابراین کاربر به صفحه ثبت نام با پیام خطا هدایت می‌شود.

7. $query = "INSERT INTO user(email,password,firstname,lastname,address,registerTime) VALUES ('$email','$password','$firstname','$lastname','$address','$registerTime')";

یک کوئری SQL برای درج اطلاعات جدید کاربر در جدول user ایجاد می‌شود.

8. if(mysqli\_query($conn,$query)){ ... } else { ... }

در صورت موفقیت آمیز بودن عملیات درج، session برای کاربر ایجاد شده و کاربر به صفحه اصلی هدایت می‌شود؛ در غیر این صورت، کاربر به صفحه ثبت نام با پیام خطا هدایت می‌شود.

با استفاده از این قطعه کد، اطلاعات جدید کاربران در پایگاه داده ذخیره می‌شود و بررسی می‌شود که آیا کاربر با این ایمیل قبلاً ثبت نام کرده است یا خیر.