

Document وب سایت خواربار فروشی

خلاصه:

در این پروژه هدف ما طراحی صفحه اول یک وبسایت خواربار فروشی است. این پروژه شامل 3 بخش می شود.

1-قسمت frontend که خود شامل دو فایل style.css و index.html می شود.

2-قسمت backend که خود شامل دو فایل php.php و php2.php می شود.

3-قسمت دیتابیس پروژه که شامل فایل products.sql می شود.

قسمت frontend :

style.css: به طور خلاصه این فایل رنگ بندی، تعیین ابعاد و تعیین موقعیت عناصر داخل صفحه نسبت به هم را به عهده دارد.

index.html: این فایل قالب بندی وب سایت ما را با استفاده از زبان html انجام می دهد. همچنین جاوا اسکریپت نوشته شده داخل تگ <script> وظیفه تغییرات در صفحه و همچنین انتقال اطلاعات به لایه backend را بر عهده دارد.

قسمت backend :

هر دو فایل php2.php و php.php ساختار یکسانی دارند با این تفاوت که اولی محصول را به سبد خرید اضافه می کند و دومی محصول را از سبد خرید حذف میکند.

قسمت database :

این فایل یک نسخه backup از دیتابیس mysql است که اطلاعات مربوط به محصولات را در خود نگه داری میکند مانند: (قیمت-نام-وضعیت سبد خرید).

توضیحات کد جاوا اسکریپت:

این قطعه کد جاوا اسکریپت عملکرد سبد خرید را برای یک برنامه وب پیاده سازی می کند. این به کاربران امکان می دهد مواردی را از سبد خرید خود اضافه یا حذف کنند و تعداد سبد را به صورت پویا به روز می کند. بیا ببینیم کد را با جزئیات بیشتری درک کنیم.

کد با بازیابی همه عناصر با نام کلاس 'cart-btn' با استفاده از تابع getElementByClassName آغاز می شود. این عناصر معمولاً دکمه‌ها یا پیوندهایی هستند که اقدامات مربوط به سبد خرید را فعال می‌کنند.

پرچمی به نام isTextChanged به false مقداردهی اولیه می شود. اگرچه این متغیر تعریف شده است، اما در قطعه کد استفاده نمی شود.

یک حلقه for روی هر پیوند جابجایی که قبلاً به دست آمده بود، تکرار می شود. هر پیوند ضامن نشان دهنده یک آیتم در برنامه وب است که می تواند اضافه یا از سبد خرید حذف شود.

با استفاده از متد `addEventListener`، شنونده رویداد به هر پیوند جابه‌جایی متصل می‌شود. این شنونده رویداد "کلیک" را ضبط می‌کند و از بروز رفتار پیش فرض جلوگیری می‌کند.

در شنونده رویداد، کد نام محصول را از ویژگی `id` عنصر کلیک شده بازیابی می‌کند. نام محصول در متغیر `productName` ذخیره می‌شود.

سپس کد کلاس "کلیک شده" عنصر کلیک شده را با استفاده از متد `classList.toggle` تغییر می‌دهد. این کلاس می‌تواند برای اهداف استایل بصری برای نشان دادن وضعیت انتخابی یک مورد استفاده شود.

کد محتوای متن عنصر کلیک شده را بررسی می‌کند. اگر متن "افزودن سبد خرید" باشد، اقدامات زیر را انجام می‌دهد:

تابع `CartCount` را با پارامتر 1 فراخوانی می‌کند تا تعداد سبد خرید را افزایش دهد.

یک شی `XMLHttpRequest (xhr)` برای رسیدگی به درخواست `HTTP POST` به فایل `'php.php'` ایجاد می‌کند. این فایل معمولاً مسئول پردازش و مدیریت عملیات مربوط به سبد خرید در سمت سرور است.

هدرهای لازم با استفاده از روش `setRequestHeader` تنظیم می‌شوند.

یک کنترل کننده رویداد `onreadystatechange` برای رسیدگی به پاسخ سرور تعریف شده است. اگر حالت آماده 4 و وضعیت 200 باشد (نشان دهنده پاسخ موفقیت آمیز)، پیامی را به کنسول ثبت می‌کند.

داده‌هایی که در درخواست ارسال می‌شوند با رمزگذاری متغیر `productName` با استفاده از تابع `encodeURIComponent` آماده می‌شوند. سپس از طریق روش ارسال ارسال می‌شود.

در نهایت، `HTML` داخلی عنصر کلیک‌شده برای نمایش «حذف» به‌روزرسانی می‌شود، که معمولاً با نماد کیسه خرید نشان داده می‌شود.

اگر محتوای متن عنصر کلیک شده «افزودن سبد خرید» نباشد، اقدامات مربوطه را برای حذف یک مورد از سبد خرید انجام می‌دهد:

تابع `CartCount` را با پارامتر 1- فراخوانی می‌کند تا تعداد سبد خرید را کاهش دهد.

یک شی `XMLHttpRequest` دیگر (`xhr`) برای رسیدگی به درخواست `HTTP POST` به فایل `'php2.php'` ایجاد می‌کند. این فایل مسئول پردازش عملیات حذف در سمت سرور است.

مشابه مورد قبلی، هدرهای لازم تنظیم شده و کنترل کننده رویداد `onreadystatechange` تعریف شده است.

داده‌های ارسالی آماده و کدگذاری می‌شوند.

درخواست ارسال می‌شود و پس از پاسخ موفقیت آمیز، پیامی به کنسول وارد می‌شود.

`HTML` داخلی عنصر کلیک‌شده برای نمایش «افزودن سبد خرید» به‌روزرسانی می‌شود که معمولاً با نماد کیسه خرید نشان داده می‌شود.

در خارج از حلقه، تابعی به نام `CartCount` تعریف شده است. این تابع یک پارامتر عدد صحیح (`int`) می‌گیرد تا نشان دهد که تعداد سبد خرید باید افزایش یا کاهش یابد.

اگر 1 `int` باشد، تعداد سبد خرید با بازیابی عنصر تعداد سبد خرید، تجزیه تعداد فعلی و به‌روزرسانی متن داخلی بر این اساس افزایش می‌یابد.

اگر 1- `int` باشد، تعداد سبد خرید با استفاده از منطق مشابه کاهش می‌یابد.

توضیحات کد `php.php` و `php2.php`:

اسکرپت PHP عملیات زیر را انجام می دهد:

با استفاده از نام سرور ارائه شده (\$servername)، نام کاربری (\$username)، رمز عبور (\$password) و نام پایگاه داده (\$dbname) با پایگاه داده MySQL ارتباط برقرار می کند. این یک سرور محلی MySQL با یک کاربر ریشه و یک رمز عبور خالی فرض می کند.

اسکرپت مقدار محصول را از آرایه \$_POST بازیابی می کند، که انتظار می رود به عنوان بخشی از درخواست HTTP POST ارسال شود.

یک دستور SQL برای به روز رسانی ستون خرید جدول محصولات در پایگاه داده آماده می کند. ردیف خاص برای به روز رسانی بر اساس ستون Name تعیین می شود که با مقدار محصول دریافتی از درخواست POST مطابقت دارد.

یک دستور آماده با استفاده از روش آماده شی اتصال پایگاه داده (\$conn) ایجاد می شود. دستور SQL به عنوان یک پارامتر ارسال می شود و از متغیرهایی برای جلوگیری از تزریق SQL استفاده می شود.

متد bind_param برای اتصال مقدار محصول به عبارت آماده شده فراخوانی می شود. نوع پارامتر به عنوان "s" مشخص می شود تا یک مقدار رشته را نشان دهد.

دستور آماده شده با استفاده از متد execute اجرا می شود. اگر اجرا با موفقیت انجام شود، اسکرپت پیغام «سفارش با موفقیت به روزرسانی شد» را خروجی می کند. در غیر این صورت، یک پیام خطا به همراه خطای خاص به دست آمده از اتصال پایگاه داده (\$conn) را خروجی می دهد.

دستور آماده شده با استفاده از روش close بسته می شود و اتصال پایگاه داده نیز با استفاده از روش close بسته می شود.

این اسکرپت PHP معمولاً به عنوان هدف XMLHttpRequest یا ارسال فرم استفاده می شود، جایی که مقدار محصول به عنوان بخشی از بار درخواست ارسال می شود. ستون خرید جدول محصولات را برای ردیفی که با نام محصول ارائه شده مطابقت دارد به 0 یا 1 به روز می کند.