

1- معرفی فایل های موجود:

:Css1.1

در این پوشه فایل های استایل تمامی صفحات قرار دارد.

:Js2.1

در این پوشه فایل های اسکریپت تمامی صفحات قرار دارد.

:Fonts3.1

در این پوشه فونت های استفاده شده در پروژه و فایل آیکون های پروژه قرار دارد.

:Img4.1

تمامی عکس های استفاده شده در پروژه در این پوشه ذخیره شده است.

:sweetAlert5.1

یک کتابخانه ی جی کوئری است برای alert های کاربر پسند و تعاملی و فایل های آن در این پوشه قرار دارد.

2- معرفی پروژه :

این پروژه فرانت اند یک وب اپلیکیشن فروشگاهی فروش شکلات است که با استفاده از: jQuery, 5 Bootstrap, JavaScript, CSS, HTML طراحی شده است.

برای پیاده سازی بک اند میتوان از زبان ها و دیتابیس های مختلفی استفاده کرد.

برای مثال میتوان با استفاده از Asp.NetCore MVC که مبتنی بر زبان C# است و دیتابیس Sql server بک اند آن را پیاده سازی کرد.

برای برقراری ارتباط با دیتابیس از EntityFramework استفاده میکند و کلاس های C# را به جداولی در دیتابیس تبدیل میکند.(روش کدفرست)

3- توابع جاوا اسکریپتی استفاده شده در این پروژه و کارکرد هر کدام:

توابع استفاده شده در هر سه صفحه :

توابع 1 تا 17 در index.js قرار دارند و توابع 18 تا 21 در questions.js قرار دارند و تابع 22 در Discounts.js قرار دارد.

1-این تابع با اسکرول خوردن صفحه اجرا میشود و کار تغییر رنگ نوبار رو انجام میدهد:

```
document.addEventListener('scroll', () => { ...
});
```

2-این تابع برای باز و بسته شدن سبد خرید است و با کلیک بر روی سبد خرید اجرا میشود:

```
var navbascket = document.querySelector('.nav-bascket').addEventListener('click', () => { ...
});
```

3-این تابع برای موجودیت سبد خرید است و ابتدا در لود صفحه اجرا میشه و سپس با هر کلیک در سبد خرید:

```
function inventory() {...}
inventory();
```

4-این تابع برای اضافه کردن محصول به سبد خرید است و با زدن دکمه افزودن به سبد خرید این تابع صدا زده میشه و آیدی محصول رو ورودیش ارسال میکنه:

```
function AddToBasket(productId) {...}
```

5-این تابع برای حذف محصول از سبد خرید است و با زدن ضربدر محصول در سبد خرید اجرا میشه:

```
function closeBox(productId) {...}
```

6-این تابع برای افزایش تعداد محصول در سبد خرید است و با زدن دکمه افزایش اجرا میشه:

```
function Increase(productId) {...}
```

7-این تابع خلاف تابع بالا عملیات کاهش محصول رو انجام میده:

```
function Decrease(productId) {...}
```

8-این تابع قیمت کل هر محصول را بدست میآورد و با تغییرات هر محصول اجرا و آیدی محصول ورودیش ارسال میشه:

```
function GetTotalPrice(productId) {...}
```

9-این تابع مبلغ قابل پرداخت را محاسبه و نمایش میده:

```
function GetPaymentAmount() {...}
```

10-این تابع مبلغ قابل پرداخت رو در فاکتور خرید نمایش میده:

```
function ShowPayment() {...}
```

11- این تابع عملیات پرداخت رو انجام میده و سبد خرید رو خالی و اطلاعات رو به حالت اول برمیگردونه و همینطور پیام موفقیت رو نمایش میده:

```
function payment() {...
```

12- این تابع محصولات موجود رو نمایش میده و هنگامی که فیلتر مورد نظر اعمال بشه اجرا میشه:

```
function Available() {...
```

13- این تابع محصولات دارای تخفیف رو نمایش میده و هنگامی که فیلتر مورد نظر اعمال بشه اجرا میشه:

```
function Discount() {...
```

14- این تابع فرمت ایمیل رو اعتبار سنجی میکنه:

```
function validateEmail(elementValue) {...
```

15- این تابع ایمیل رو ثبت و پیام موفقیت رو نمایش میده:

```
function emailSend() {...
```

16- این تابع عملیات فیلتر قیمت ها رو انجام میده و با انتخاب فیلتر مورد نظر اجرا میشه:

```
function PriceFilter() {...
```

17- این تابع کار امتیاز دهی به محصولات رو انجام میده و هنگام انتخاب ستاره ها اجرا میشه:

```
function Score(starId) {...
```

18- پس از لود صفحه سوالات و با گذشت 5 ثانیه این تابع اجرا و باکس پشتیبانی را نمایش میدهد:

```
setTimeout(function () {...
```

19- این تابع برای بستن باکس پشتیبانی است و با زدن ضربدر باکس پشتیبانی اجرا میشود:

```
var btnClose = document.querySelector('.btn-close').addEventListener('click', () => {...
```

20- این تابع اطلاعات باکس پشتیبانی را اعتبار سنجی سپس ارسال میکند و با زدن دکمه ثبت باکس پشتیبانی اجرا میشود:

```
function sendInformation() { ...  
}
```

21- عملیات بسته شدن باکس پشتیبانی رو انجام میدهد ولی بعد از ثبت اطلاعات اجرا میشود:

```
function HideBoxSupporter(){ ...  
}
```

22- این تابع کار تایمر محصولات دارای تخفیف رو انجام میدهد و پس از لود صفحه تخفیفات اجرا میشود:

```
var times= ['#Discount-timer-1','#Discount-timer-2','#Discount-timer-3','#Discount-timer-4'];  
times.forEach(element => { ...  
});
```