CAN302 – Technologies for e-commerce

Division of communication and network
Xi’an Jiaotong-Liverpool University

Lab 6 – Preparing the SKU for Shopping-cart

Aim

The aim of this lab is to learn how to monitor the options chosen by customers and make the SKU be ready for shopping-cart.

Tips:

1. If you are not sure why you are doing something, ask a TA. This is what they are here for.

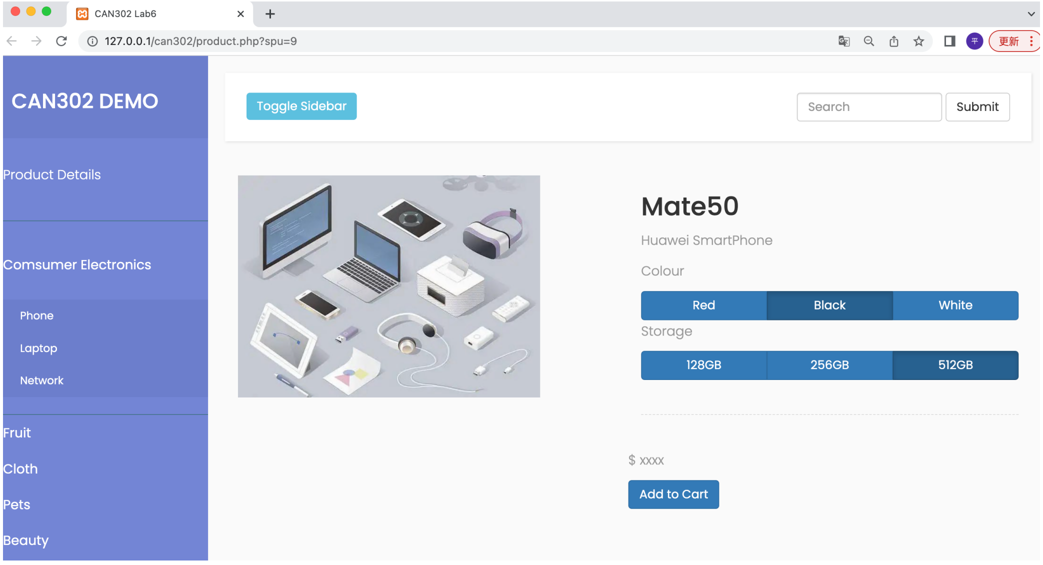
2. The M-Dev-Store online videos are good references while our labs have different focus. If you want to be an expert, you are recommended take both labs and on-line videos.

3. The forums @ LMO are available for questions and discussions.

4. These labs are expected take more than the 2 allocated hours. You should complete them in your own time before the next lab. Practice makes perfect!

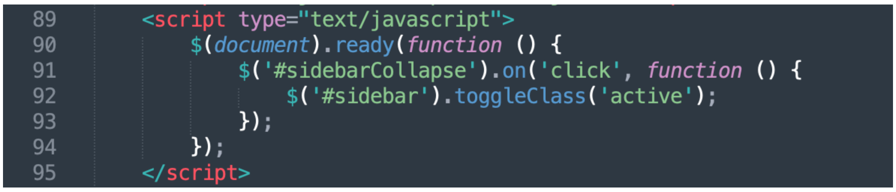
Monitor the webpage by JS:

1. Till last lab, we can show the choices to customers like following:



It is only a “dummy” to show the effect of click but nothing happens when customers make their choices. This lab we would improve the monitor function to make the SKU be ready for the shopping-cart.

To do the monitor, we need to use JS function. Actually, the toggle button to close/open the sider-bar based on JS monitor. The code is as following:

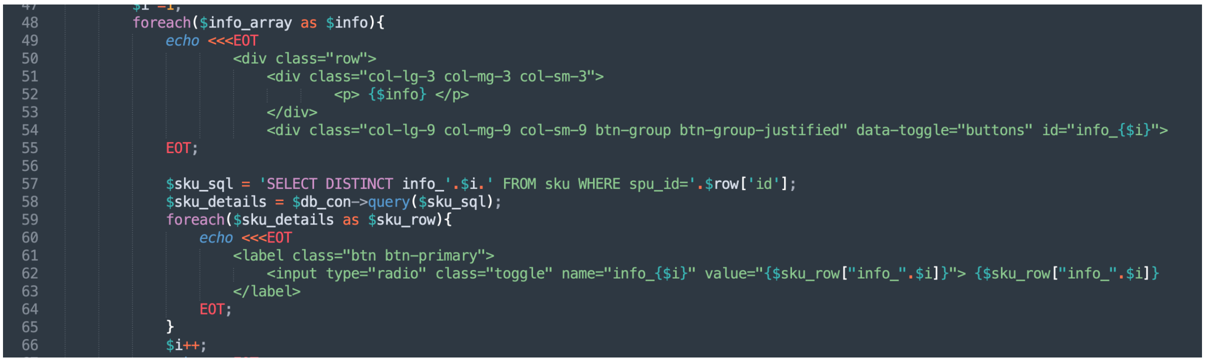


```
    <script type="text/javascript">
        $(document).ready(function () {
            $('#sidebarCollapse').on('click', function () {
                $('#sidebar').toggleClass('active');
            });
        });
    </script>
```

In which, document mean the whole webpage and ready means when all DOMs are loaded. JS has many functions to monitor the actions by the user to the webpage. It is like the computer to receive the keyboard and mouse action. To receive the action, a target should be specified. It can be an element id or a class name. The above sample code would monitor the mouse “click” on the element which id is “sidebarCollapse”. Within the function, the element which id is “sidebar” would toggle between active and inactive. And the “#” told JS the variable is an element id.

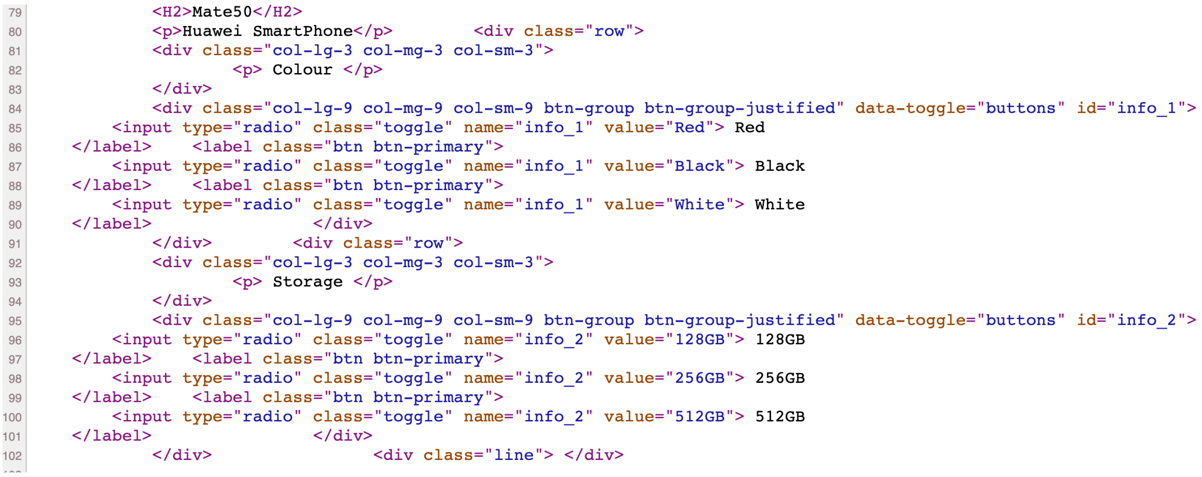
Monitor the options by JS:

2. In our case, we need to monitor the selections by the customer. Therefore, we need to specify the id to the btn-group and put values to each option. The PHP code would be improved as:



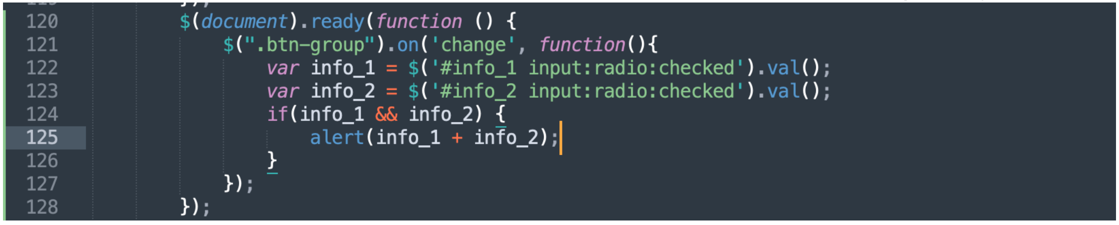
```
        $i =1;
        foreach($info_array as $info){
            echo <<<EOT
                <div class="row">
                    <div class="col-lg-3 col-mg-3 col-sm-3">
                            <p> {$info} </p>
                    </div>
                    <div class="col-lg-9 col-mg-9 col-sm-9 btn-group btn-group-justified" data-toggle="buttons" id="info_{$i}">
            EOT;

            $sku_sql = 'SELECT DISTINCT info_'.$i.' FROM sku WHERE spu_id='.$row['id'];
            $sku_details = $db_con->query($sku_sql);
            foreach($sku_details as $sku_row){
                echo <<<EOT
                    <label class="btn btn-primary">
                        <input type="radio" class="toggle" name="info_{$i}" value="{$sku_row["info_".$i]}"> {$sku_row["info_".$i]}
                    </label>
                EOT;
            }
            $i++;
```

The output HTML code is as following:



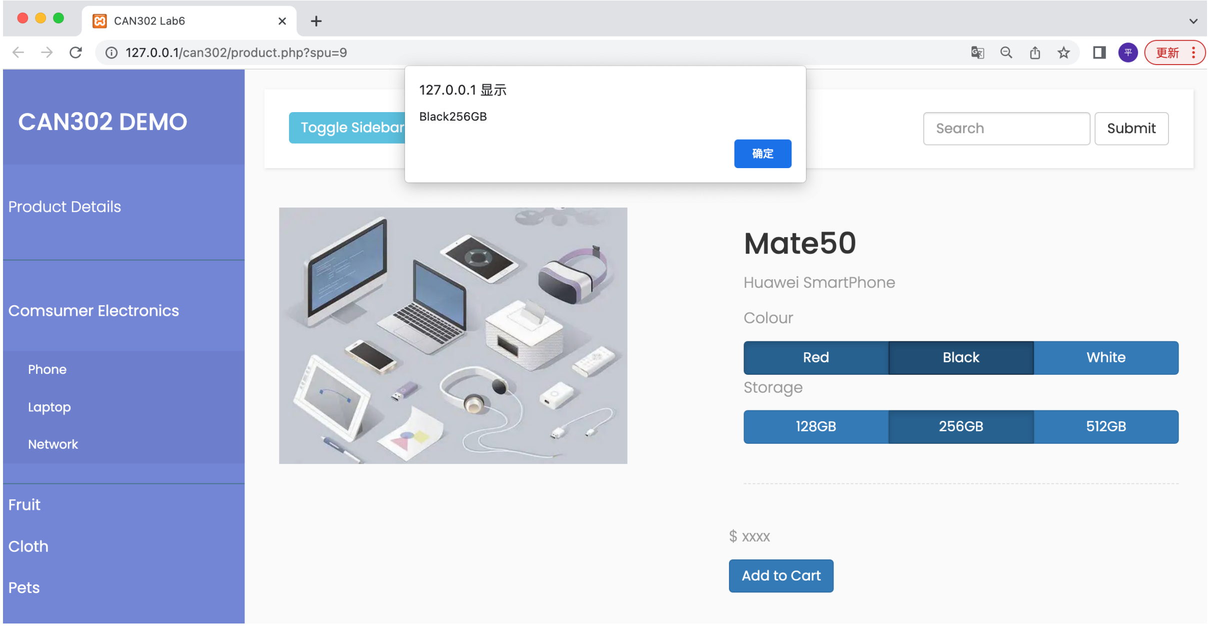
```
            <H2>Mate50</H2>
            <p>Huawei SmartPhone</p>        <div class="row">
            <div class="col-lg-3 col-mg-3 col-sm-3">
                    <p> Colour </p>
            </div>
            <div class="col-lg-9 col-mg-9 col-sm-9 btn-group btn-group-justified" data-toggle="buttons" id="info_1">
        <input type="radio" class="toggle" name="info_1" value="Red"> Red
    </label>    <label class="btn btn-primary">
        <input type="radio" class="toggle" name="info_1" value="Black"> Black
    </label>    <label class="btn btn-primary">
        <input type="radio" class="toggle" name="info_1" value="White"> White
    </label>                </div>
            </div>        <div class="row">
            <div class="col-lg-3 col-mg-3 col-sm-3">
                    <p> Storage </p>
            </div>
            <div class="col-lg-9 col-mg-9 col-sm-9 btn-group btn-group-justified" data-toggle="buttons" id="info_2">
        <input type="radio" class="toggle" name="info_2" value="128GB"> 128GB
    </label>    <label class="btn btn-primary">
        <input type="radio" class="toggle" name="info_2" value="256GB"> 256GB
    </label>    <label class="btn btn-primary">
        <input type="radio" class="toggle" name="info_2" value="512GB"> 512GB
    </label>                </div>
            </div>                <div class="line"> </div>
```

We have more than one element need to be monitored. So we can monitor the class “btn-group”. In JS, “.” is used to identify a class as you have seen many samples in CSS file. We can use the following code to monitor the selections:



```
        $(document).ready(function () {
            $(".btn-group").on('change', function(){
                var info_1 = $('#info_1 input:radio:checked').val();
                var info_2 = $('#info_2 input:radio:checked').val();
                if(info_1 && info_2) {
                    alert(info_1 + info_2);
                }
            });
        });
```

Here, the alert is used to show the selected choices. The effect on the webpage is as following:



127.0.0.1 显示

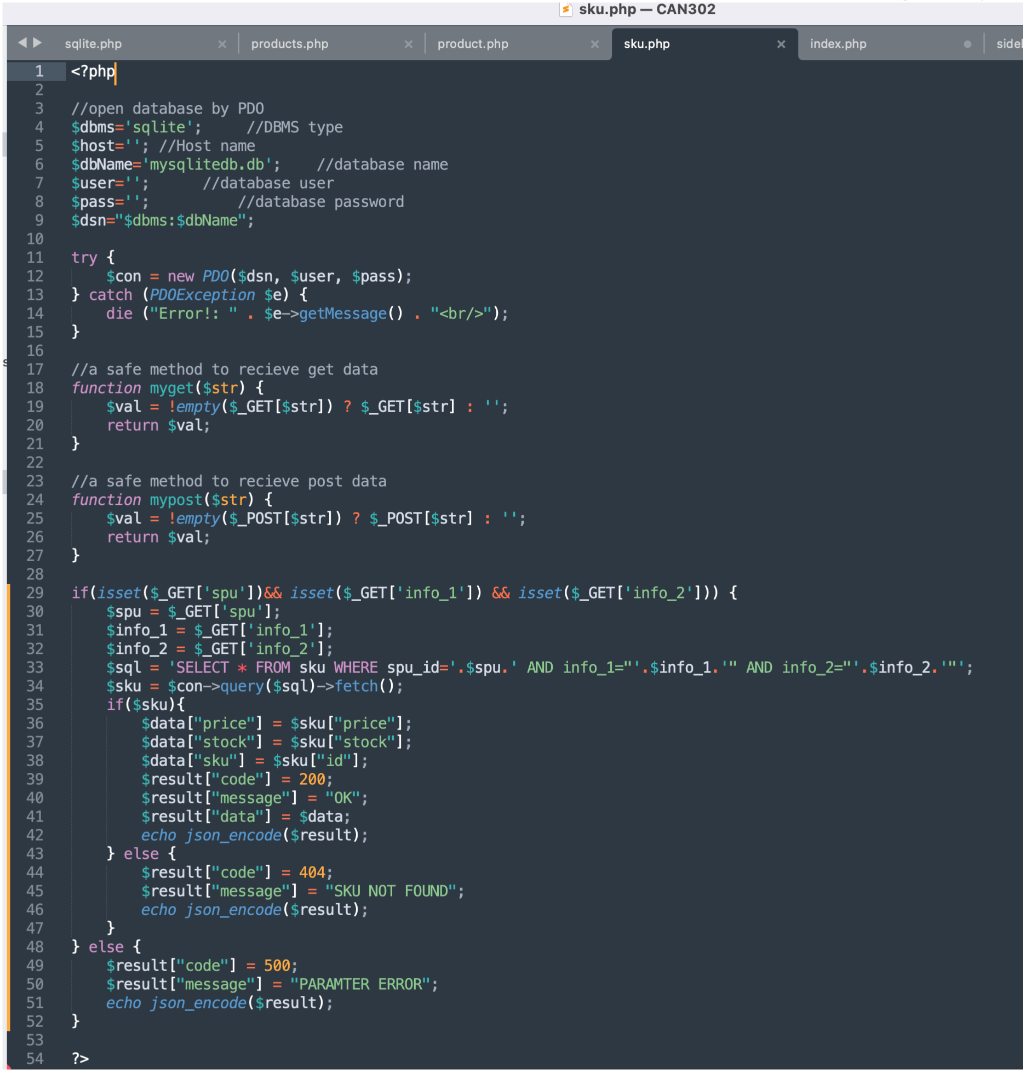
Black256GB

Please the monitor here is “change” but not “click”. You may try the difference by yourself.

GET the SKU information:

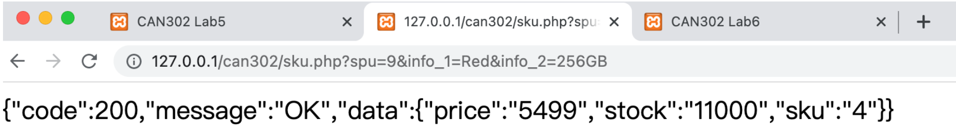
3. SKU means different combinations. Once the combination is set, we need to know which SKU it is, and change the price, image … accordingly. It is not a clever way to load the whole webpage. The JS code can update the information through the AJAX request. JSON is a popular format for the interface of web request. We need to define the API and then code the service (back-end) and JS part (front-end) separately.

   The back-end code is as:



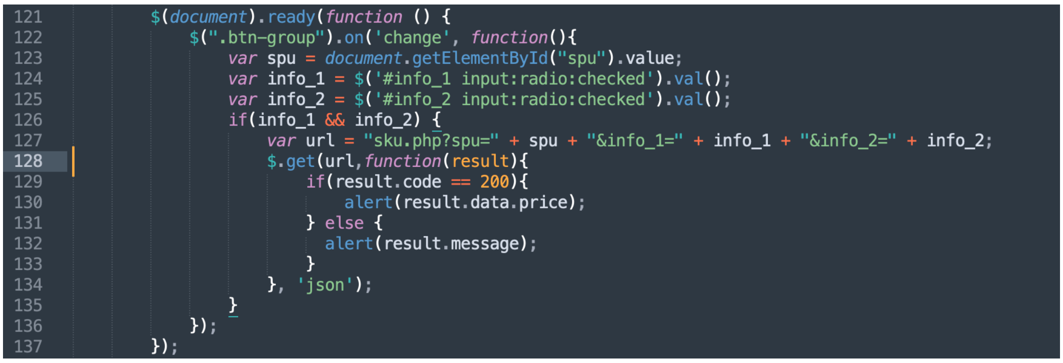
```php
<?php

//open database by PDO
$dbms='sqlite';     //DBMS type
$host=''; //Host name
$dbName='mysqlitedb.db';    //database name
$user='';      //database user
$pass='';          //database password
$dsn="$dbms:$dbName";

try {
    $con = new PDO($dsn, $user, $pass);
} catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}

//a safe method to recieve get data
function myget($str) {
    $val = !empty($_GET[$str]) ? $_GET[$str] : '';
    return $val;
}

//a safe method to recieve post data
function mypost($str) {
    $val = !empty($_POST[$str]) ? $_POST[$str] : '';
    return $val;
}

if(isset($_GET['spu'])&& isset($_GET['info_1']) && isset($_GET['info_2'])) {
    $spu = $_GET['spu'];
    $info_1 = $_GET['info_1'];
    $info_2 = $_GET['info_2'];
    $sql = 'SELECT * FROM sku WHERE spu_id='.$spu.' AND info_1="'.$info_1.'" AND info_2="'.$info_2.'"';
    $sku = $con->query($sql)->fetch();
    if($sku){
        $data["price"] = $sku["price"];
        $data["stock"] = $sku["stock"];
        $data["sku"] = $sku["id"];
        $result["code"] = 200;
        $result["message"] = "OK";
        $result["data"] = $data;
        echo json_encode($result);
    } else {
        $result["code"] = 404;
        $result["message"] = "SKU NOT FOUND";
        echo json_encode($result);
    }
} else {
    $result["code"] = 500;
    $result["message"] = "PARAMTER ERROR";
    echo json_encode($result);
}

?>
```

The demo use GET parameters. The data can be prepared in array format then use the json_encode function to output the JSON format. And please **BE CAREFUL**, the demo code here is a **DIRTY** one. For complex API, POSTMAN is a popular test tool. we can directly visite this simple API by browser as:


CAN302 Lab5
127.0.0.1/can302/sku.php?spu
CAN302 Lab6
127.0.0.1/can302/sku.php?spu=9&info_1=Red&info_2=256GB

```
{"code":200,"message":"OK","data":{"price":"5499","stock":"11000","sku":"4"}}
```

Then we can update the JS as:



```javascript
        $(document).ready(function () {
            $(".btn-group").on('change', function(){
                var spu = document.getElementById("spu").value;
                var info_1 = $('#info_1 input:radio:checked').val();
                var info_2 = $('#info_2 input:radio:checked').val();
                if(info_1 && info_2) {
                    var url = "sku.php?spu=" + spu + "&info_1=" + info_1 + "&info_2=" + info_2;
                    $.get(url,function(result){
                        if(result.code == 200){
                            alert(result.data.price);
                        } else {
                          alert(result.message);
                        }
                    }, 'json');
                }
            });
        });
```

“$.get” is the JS AJAX method to trigger the GET request. The result would be received and then can be processed. For details of this function, you can Google or Bing it.

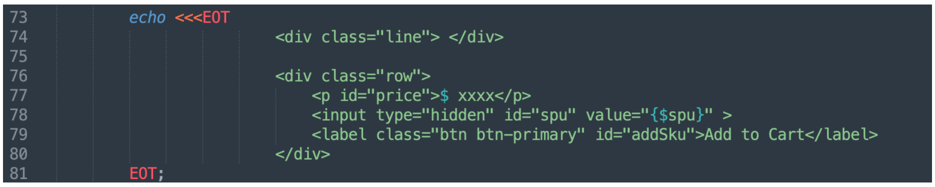
Please notice the “spu” parameter is need for the SKU API. How to get the “spu id” by JS? Please figure it out.

Till now, this function is to alert the price when there is the combination and alert the message when there is no such combination. You may try to see the result by yourself.

Then we need to update the rest of the webpage and make it ready to add a product to the shopping-cart by the user.

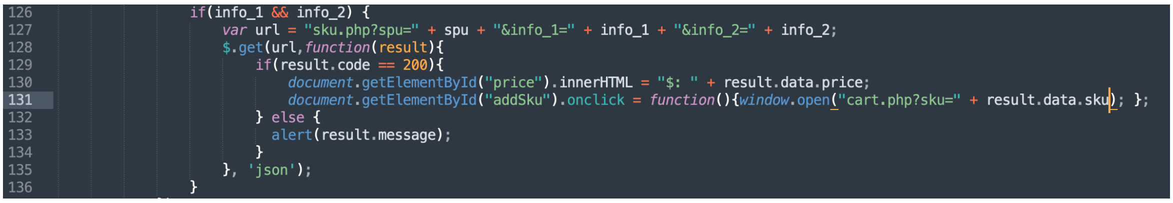
Update the price and make the form ready!

4. It is time to use the data from SKU API. We need to set the id for price and submit elements. The code is as:



```
        echo <<<EOT
                    <div class="line"> </div>

                    <div class="row">
                        <p id="price">$ xxxx</p>
                        <input type="hidden" id="spu" value="{$spu}" >
                        <label class="btn btn-primary" id="addSku">Add to Cart</label>
                    </div>
        EOT;
```

Then, we can update them in the JS function as:



```
                if(info_1 && info_2) {
                    var url = "sku.php?spu=" + spu + "&info_1=" + info_1 + "&info_2=" + info_2;
                    $.get(url,function(result){
                        if(result.code == 200){
                            document.getElementById("price").innerHTML = "$: " + result.data.price;
                            document.getElementById("addSku").onclick = function(){window.open("cart.php?sku=" + result.data.sku); };
                        } else {
                          alert(result.message);
                        }
                    }, 'json');
                }
```

Once the user select the combinations, the price would be updated and click the “Add to cart” button would open the “cart.php” with sku parameter. We would make the cart part in next lab.

Home work

5. There are **DIRTY** code and some interaction “bugs” within the code.

Interaction bugs example:

- There is the alert for wrong combinations, but the wrong choice was not canceled.

Dirty code example:

- The above code CANNOT work for the laptop product. It has 3 different properties.
- It is better to disable the impossible choices rather than alert the wrong combination.

Please try to fix such bugs and make the code **CLEAN**.