NANYANG TECHNOLOGICAL UNIVERSITY

CZ3006

Net-Centric Computing

Assignment 2

Written Report

Written By: Mock Jin Wei (1721242F)

Tutorial Group: TS5

Submission Date: 18 April 2018

Report Content

1. HTML Document

a) Textboxes to collect User's Name, Number of Apples, Oranges and Bananas Ordered.

b) JavaScript Validation for validity of inputs and the displaying of relevant error messages for invalid inputs.

c) Textbox that display the Total Cost of User's Order.

d) Radio Buttons that display Payment Methods.

e) Submit Button to send the order to the web server.

2. Server-Side PHP Program

a) Returning a HTML document (Order Receipt).

b) Order.txt file will be updated with the latest Total Amount of Apples, Oranges and Bananas ordered from all Users.

3. Additional Features

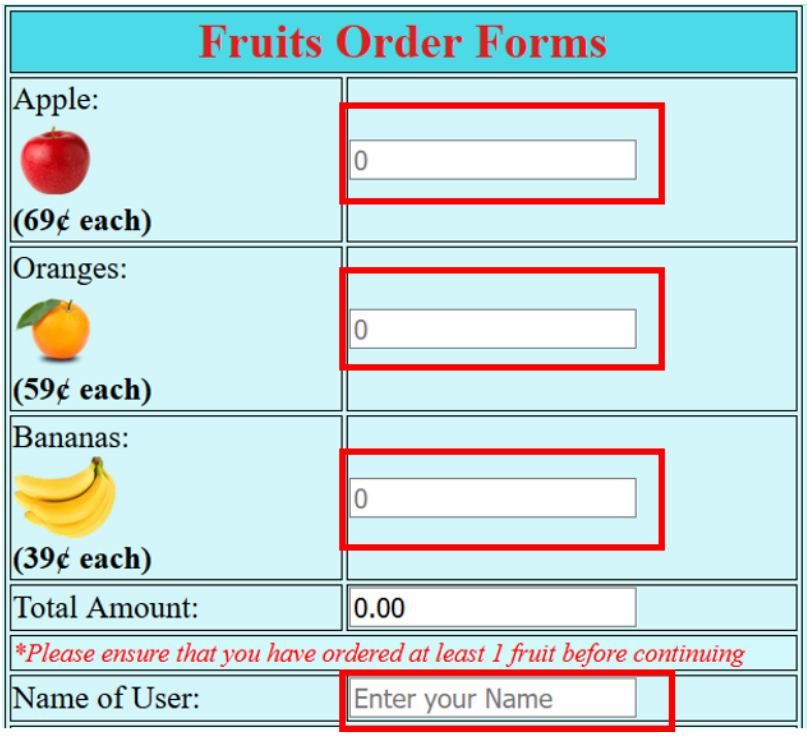
4. Listing of Files

Extra Notes:

- *All red boxes that surrounds a particular part of an image are the key focus for that image.*
- *All subparts (E.g. 1a, 1b, 1c….) begins on a new page, with the exception of 1d, 1e and 2a.*
- *The term "Ordering Textboxes" is used frequently here. It refers to the textboxes that allows the Users to enter the Quantity of the particular fruits that they wish to order.*
- *"Total Amount" textbox refers to the textbox that display the Order Total Amount.*

1. HTML Document

1a) Textboxes to collect User's Name, Number of Apples, Oranges and Bananas Ordered.

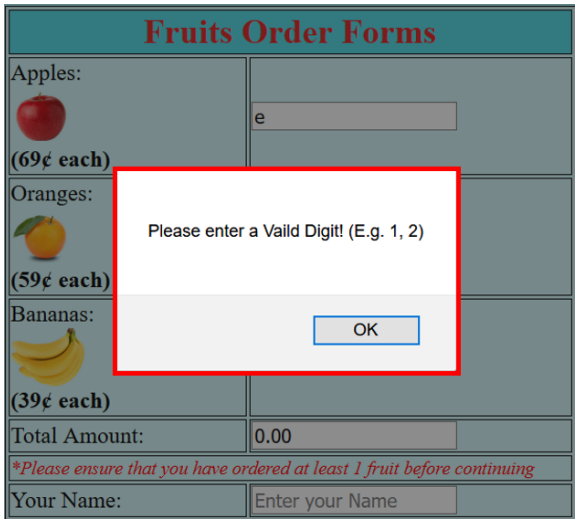


Img 1

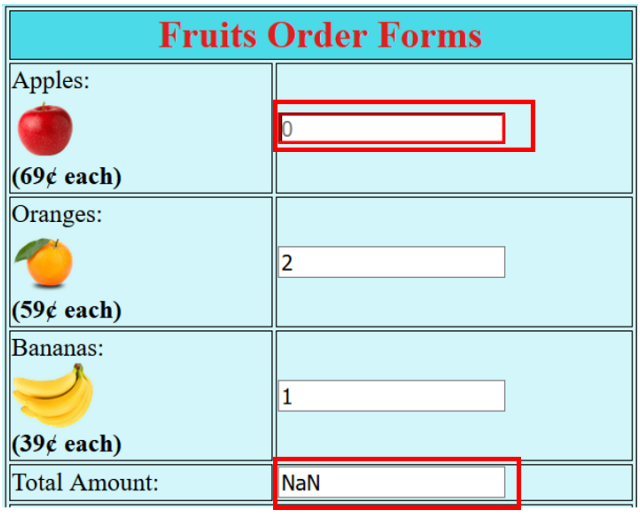
As shown in Img 1, a total of 4 textboxes, which is highlighted in red, will be provided for the User to enter the corresponding values for their order. User will have to at least order a fruit (1 of the 3 fruit Ordering Textbox should contains a valid digit) and should enter their Name before they can submit the order form to the web server.

To ensure that the User will submit valid inputs value to the web server, various "Hint" text will appear in the textboxes before the User enter any texts. These "Hint" text will inform the User on what is the valid inputs value that is expected from them.

1b) JavaScript Validation for validity of inputs and the displaying of relevant error messages for invalid inputs.



Img 2



Img 3

When the User has entered an invalid input (E.g. Alphabets or Symbols) into the Ordering Textboxes, an error pop-up message as shown in Img 2 will be displayed. The purpose of the error pop-up message is to warn the User about their invalid input and request them to change their input into a valid one. The textbox that contains the invalid input will also get highlighted in red *(Seen in Img 3)* to bring the attention of the User to that particular textbox.

If the User had entered an invalid input in any 1 of the 3 Ordering Textboxes, the "Total Amount" textbox for the Order will be changed to "NaN" *(Seen in Img 3)*. This is an indication that the Total Amount of the Order cannot be calculated and the User will have to entered valid inputs in the 3 Ordering Textboxes before the Total Amount can be calculated.

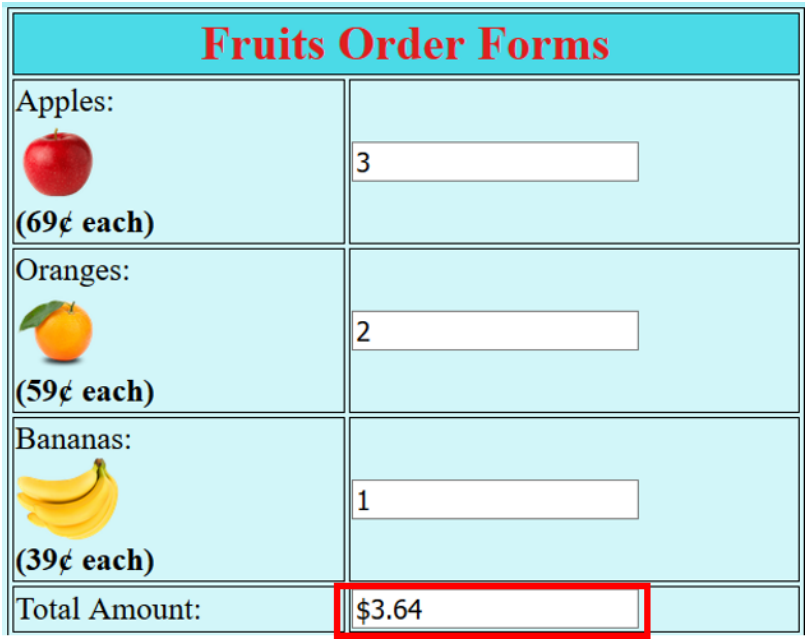
The "Total Amount" textbox will be updated with the value "NaN" and the corresponding invalid input textbox will get highlighted in red after the User had clicked on the "OK" button in the error pop-up message box.

Below is a short code snippet to show how validation of valid digit is achieved in JavaScript:

```
// Function to validate if the entered text is a digit
function validateValue()
{

var vApple, vOrange, vBanana; // Variables to store data for subsequent access
var cApple, cOrnages, cBanana; //Variables to store the results from validating if the entered digit is a vaild one
var costA = 0.69; // Declaring the cost of Apple
var costO = 0.59; // Declaring the cost of Orange
var costB = 0.39; // Declaring the cost of Bananas

vApple = document.getElementById("appleText").value; // Assigning the value in the Apple textbox to vApple
vOrange = document.getElementById("orangeText").value; // Assigning the value in the Orange textbox to vOrange
vBanana = document.getElementById("baText").value; // Assigning the value in the Banana textbox to vBanana

//Next 3 lines are formula that is used to check if the entered values in the respective textboxes are Whole Number Digits
cApple = vApple % 1;
cOrange = vOrange % 1;
cBanana = vBanana % 1;

//If-else statements that display different results based on whether the entered value is a Valid Digit
        if (isNaN(vApple) || cApple != 0) {
                alert("Please enter a Vaild Digit! (E.g. 1, 2)");
                document.getElementById("appleText").value = "";
         }
```

**Code snippets show validation for ordering of Apples only. Similar codes are used to validate ordering of Oranges and Bananas.*

1c) Textbox that display the Total Cost of User's Order



Img 4

When the User entered valid digits in the Ordering Textboxes, the value inside the "Total Amount" textbox will be updated to reflect the latest Total Cost of the whole Order *(Shown in Img 4)*. The value in the textbox will be recalculated immediately after the User had make changes to the value in the Ordering Textboxes. For example, if the User has decided to change the value in the Apple Ordering Textbox to 4 instead of 3, the value inside the "Total Amount" textbox will be recalculated and a new value of "4.33" will be displayed.

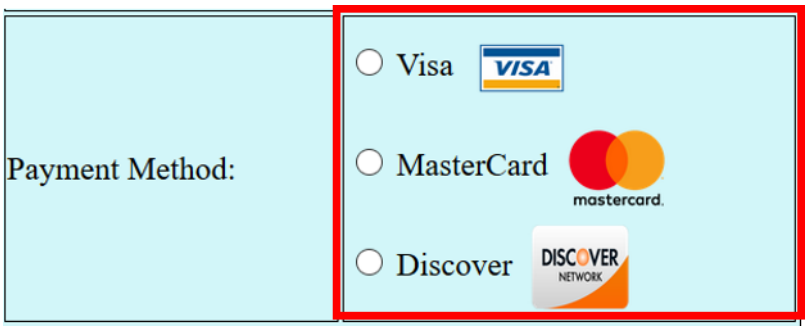
In order to achieve the calculation of total amount ordered, the following codes is used:

```
//Once the Values has passed all vaildation, counting of the total amount will happen here
else {
    var totalACost = parseFloat(vApple * costA); //Counting Total Cost of Apple
    var totalOCost = parseFloat(vOrange * costO); //Counting Total Cost of Oranges
    var totalBCost = parseFloat(vBanana * costB); //Counting Total Cost of Bananas
    var totalAmt = totalACost + totalOCost + totalBCost; //Counting Total Cost of all the Fruits

    document.getElementById("totalText").value = totalAmt.toFixed(2); //Displaying the Total Cost, rounded to 2 decimal points.
}
```

The above code snippet shows the calculation of the Total Cost for the individual fruits items, as well as the calculation for the Order's Total Amount. Once the Order's Total Amount is calculated, the result will be displayed in the "Total Amount" textbox.

Together with the code snippets that is shown in 1b), the function validateValue() is constructed. This function will enable the webpage to validate all the User inputs and accurately calculate the Total Amount ordered from the user given values.

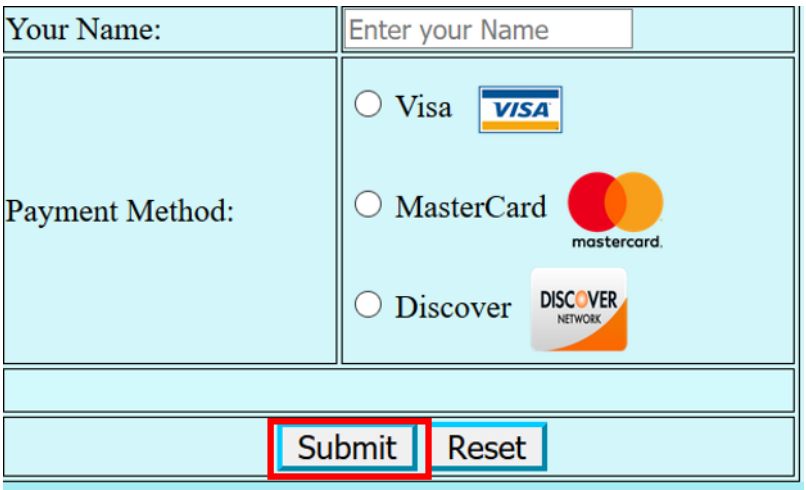
1d) Radio Buttons that display Payment Methods



Img 5

The User will have to choose between 1 of the 3 given Payment Method listed in the webpage to pay for their order. Img 5 shows the design layout for the User to choose their Payment Method.

1e) Submit Button to send the order to the web server.



Img 6

Once the User has finish ordering their fruits, they will click on the "Submit" button *(Shown in Img 6)* to submit their Order to the web server. However, before the webpage actually submits the Order to the server, a series of validations will take place to ensure that the User had entered all the required inputs.

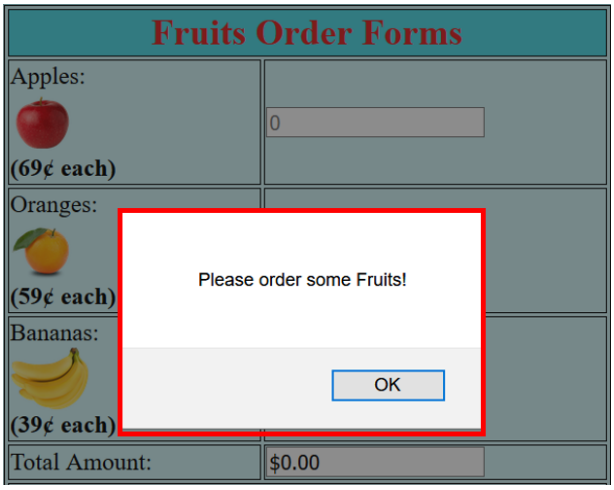
The validation will includes checking if the User had ordered at least 1 fruit, ensuring that the User had entered their Name when they submit the Order and the User had selected a Payment Method. The code snippets in the next page will show how this type of validation is done.

```
function validateEmpty() {

var eName, eTotal, ePay; //Variables to store the Name, Total Amount Value and Payment Method from the HTML
eName = document.getElementById("nameText").value;
eTotal = document.getElementById("totalText").value;
ePay = document.getElementsByName("paymethod");

//If statement to check if the User has entered a Name
if(eName == "" || eName == "Enter your Name")
{
            alert("Please enter your name!");
            eName.value = "Enter your Name";
            document.getElementById("nameText").style.borderColor = "red";
            return false;

}

//If statement to check whether the User has ordered at least 1 fruit
if(eTotal == "$0.00" || eTotal == "NaN" )
{
            alert("Please order some fruits!");
            return false;

}

//For Statement to check if the User has selected a Payment Method
for(var i = 0, len = ePay.length; i<len; i++)
{
        if(ePay[i].checked)
        {
                return true;
        }
}
            alert("Please select a Payment Method!");
            return false;
```

As seen from the above code snippet for the function validateEmpty(), the webpage will check if the User had left the “Username” textbox empty and will check if they have at least ordered 1 valid fruits, by making sure that the Total Amount is not 0 or NaN. The webpage will also check if the User has selected a Payment Method before submitting the Order.

In an event that the User had not entered their Name or there is an invalid fruit ordered, the following error pop-up messages in Img 7 and 8 will be shown to warn the User about the respectively errors. A similar error will be shown if the User has not select a Payment Method.

Img 7



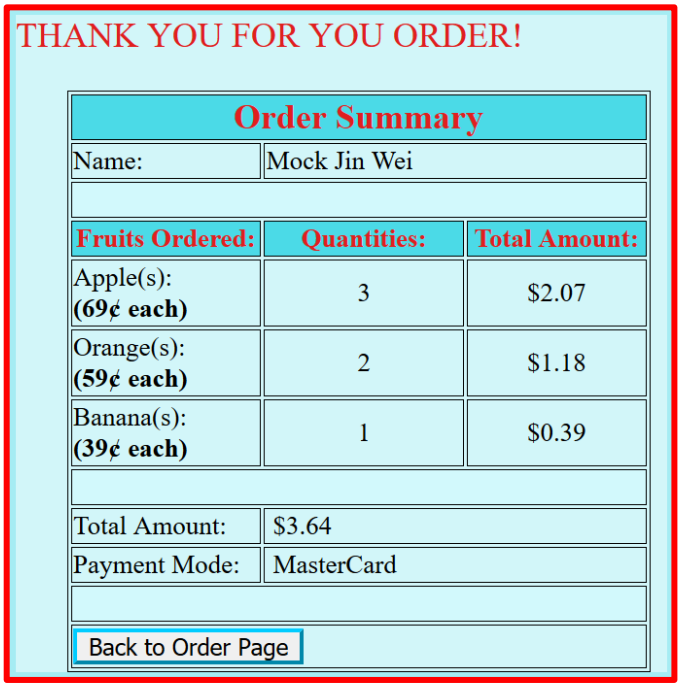
Img 8

Once the Order had cleared the validations, the Order will be sent to the web server. This is done by the following code snippet, where the form action is specify to send the Order to the PHP program for processing.

```
<form action="orderReceipt.php" method="post" onsubmit="return validateEmpty()">
```

2. Server-Side PHP Program

2a) Returning a HTML document (Order Receipt) to the User



THANK YOU FOR YOU ORDER!

Order Summary		
Name:	Mock Jin Wei	
Fruits Ordered:	**Quantities:**	**Total Amount:**
Apple(s): **(69¢ each)**	3	$2.07
Orange(s): **(59¢ each)**	2	$1.18
Banana(s): **(39¢ each)**	1	$0.39
Total Amount:	$3.64	
Payment Mode:	MasterCard	
Back to Order Page		

Img 9

Once the Order had been sent to server side for processing, the PHP program will extracts all the relevant inputs such as the Name, Number of Apples Ordered, Payment Method and etc., out from the Ordered Form. With the extracted inputs, the PHP program will calculate the Total Order Cost for the User. After all the extracting and calculating, the PHP program will return an HTML document to the User as a Receipt. The receipt will contains information such as the User Name, Total Amount of Apple, Oranges and Bananas ordered, Total Amount Ordered for that Particular Fruit, Order Total Amount and the chosen Payment Method. All these data information will be displayed in a Table and Img 9 shows an example of an Order Receipt that is returned from the PHP program after it has process all the inputs.

The code snippets shown in the next page will show how the PHP program process its inputs when it receives the Order from the HTML document.

```php
<?php
	$name =  $_POST["Uname"]; //Getting the value in the textbox named "Uname" in the HTML Document
	$apple =  $_POST["apple"]; //Getting the value in the textbox named "apple" in the HTML Document
	$orange =  $_POST["orange"]; //Getting the value in the textbox named "orange" in the HTML Document
	$banana =  $_POST["banana"]; //Getting the value in the textbox named "banana" in the HTML Document

	$Capple = $apple * 0.69; //Calculating the total cost Apples
	$Corange = $orange * 0.59; //Calculating the total cost Oranges
	$Cbanana = $banana * 0.39; //Calculating the total cost Bananas
	$total =  $Capple + $Corange + $Cbanana; //Calculating the total cost of the whole order

	$pay =  $_POST["paymethod"]; //Getting the value in the textbox named "paymethod" in the HTML Document
```

*Above code snippet shows how the PHP program extracts the inputs from the Order and how it calculates the Total Costs from the extracted inputs.

```
<!-- Row to Display Number of Apples Ordered -->
 <tr>
      <td>Apple(s): <br/> <b>(69&#162; each) </b></td>
      <td class = "cellData"><?php print ("$apple"); ?></td>
      <td class = "cellData"> $<?php print ("$Capple"); ?></td>
 </tr>
```

*Above code snippet shows how the PHP program will display the processed data in a Table format when it returns an Order Receipt HTML document back to the User. For the displaying of other data like the Name and Payment Method, similar codes are used to achieve it.

2b) Order.txt file will be updated with the latest Total Amount of Apples, Oranges and Bananas ordered from all Users.

```
Total Number of Apple(s): 3
Total Number of Orange(s): 2
Total Number of Banana(s): 1
```

Besides returning a HTML document, the PHP program will also maintains a record of the Total Numbers of Apples, Oranges and Bananas ordered from all Users. This record, which comes in a form of a txt fie (order.txt), will be updated whenever there are new Orders received from the webpage.

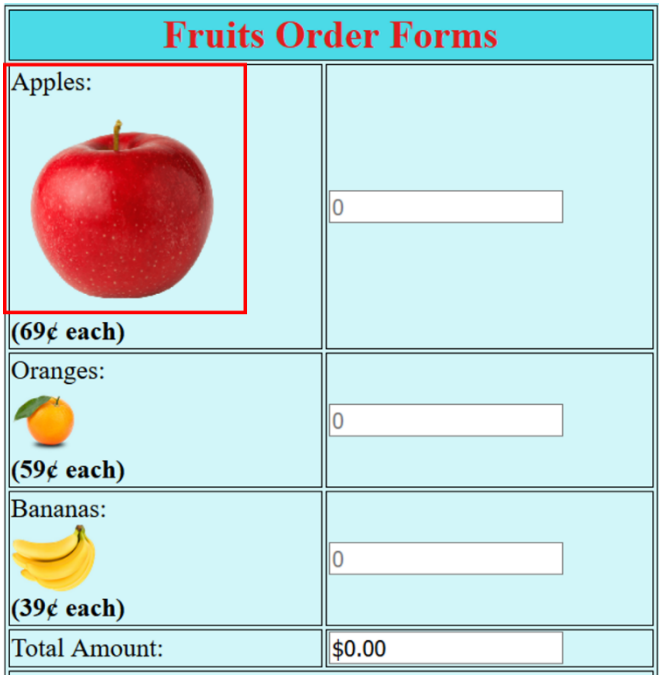
To update the order.txt file, the following codes were used:

```
$totalOrder = array(); //Declaring an array to store the total amount ordered for apples, oranges and bananas

$dbseparator = ':';

$orderfile = fopen("order.txt", "r") or die("Unable to open file!"); //Open order.txt file for reading the total amount in each fruits

        for($i =0; $i<3; $i++){
                        $value = fgets($orderfile); // Read the lines from file
                        $dblinepos = strpos($value, $dbseparator); //Find the positon of the value
                        $value = substr($value, $dblinepos+2); //Get the substring
                        $totalOrder[$i] = (int)$value; //Assigning the substring value after changing into intger
                }

$totalApples = $totalOrder[0] + $apple; //Setting position 0 of totalOrder Array to display the total amount of Apple ordered
$totalOranges = $totalOrder[1] + $orange; //Setting position 1 of totalOrder Array to display the total amount of Orange ordered
$totalBananas = $totalOrder[2] + $banana; //Setting position 2 of totalOrder Array to display the total amount of Banana ordered

fclose($file); //Close order.txt file

$orderfile = fopen("order.txt", "w") or die("Unable to open file!"); //Open the order.txt file for updating the total amount in each fruits

$Tapple = "Total Number of Apple(s): $totalApples \n" ; //Formatting how the data should be displayed in order.txt file
fwrite($orderfile, $Tapple); //Updating the total amount of Apples ordered from all User

$Torange = "Total Number of Orange(s): $totalOranges  \n" ; //Formatting how the data should be displayed in order.txt file
fwrite($orderfile, $Torange); //Updating the total amount of Oranges ordered from all User

$Tbanana = "Total Number of Banana(s): $totalBananas \n" ; //Formatting how the data should be displayed in order.txt file
fwrite($orderfile, $Tbanana); //Updating the total amount of Bananas ordered from all User

fclose($file); //Close order.txt file
```

With the help from a series of PHP codes, the order.txt file will be opened once the PHP program receives the Order from the HTML document. All current total amount of Apples, Oranges and Bananas in order.txt file will get extracted and will be stored in an Array. When the PHP program has finish extracting the inputs from the Order, the program will update the Total Number of Apples, Oranges and Bananas by adding the extracted inputs values with the corresponding fruits values that is stored in the Array. A new total amount of the respectively fruits will be generated as a result and the latest total amount will get written back to order.txt file before the PHP program ends its process.

3) Additional Features

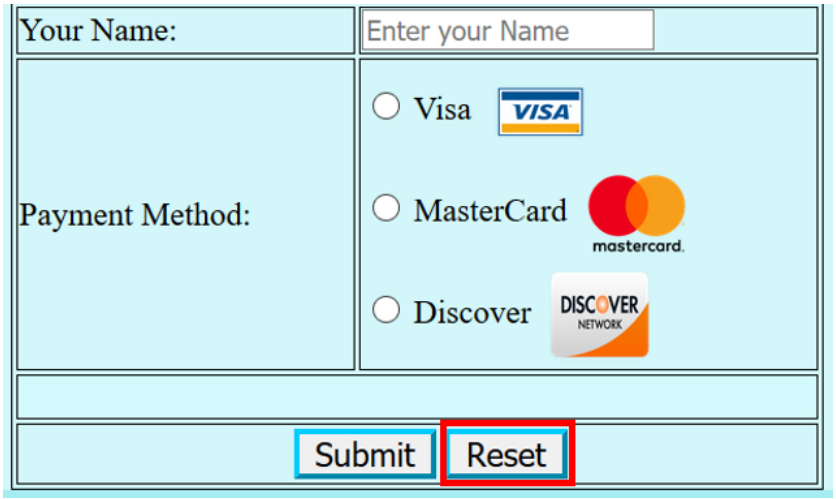
3a) Auto Enlargement of Fruits Image



Img 10

Whenever the User hover their mouse over a particular fruit image, the webpage will automatically enlarge the size of the fruit image *(Shown in Img 10)*, making it easier for the User to see the fruits image.

3b) Reset Button to reset all entered inputs



Img 11

To bring greater convenience to the User, if they wish to re-enter all their inputs in the textboxes, they can simply click on the “Reset” button and all the textboxes will get reset back to their defaults values. By doing so, the User will not need to manually clear off the current inputs in the textboxes and can instead, just enter their new inputs.

4) Listing of Files

The following files are the source files that are part of this Assignment

1. orderForm.HTML
    - HTML document for Part 1 of the Assignment
2. orderReceipt.PHP
    - PHP program file for Part 2 of the Assignment
3. order.txt
    - txt file for recording of Total Amount of Apples, Oranges and Bananas
4. Images
    - Contains all relevant images for the Assignment