4/11/2019

CZ3006 NET CENTRIC COMPUTING

Assignment II – Web Application using JavaScript and PHP

LEOW ROU SHAN
LAB GROUP TS1, U1720758J

Table of Contents

Assignment Summary 2
1. HTML Document 3
1.1. User's Name Input 3
1.2. Apples/Oranges/Bananas Order 4
1.3. Total Cost 5
1.4. Payment Method 6
1.5. Submit Order 7
2. PHP Document 8
3. Source Code 10
fruitOrder.html 10
style.css 12
submit.php 13

Assignment Summary:

1. Write an HTML document to create a form that contains:
    a. A text box to collect the user's name.
    b. Three text boxes for the user to key in the numbers of apples (69 cents each), oranges (59 cents each), and bananas (39 cents each) to purchase respectively. Whenever the user's input in any of these text boxes changes, the validity of the input is checked at the client side using JavaScript. A valid input is defined as a sequence of one or more digits. If the input is not valid, an alert message is produced, and the user is asked to input again.
    c. A text box showing the total cost of the user's order. The purpose of this text box is to show the total cost of order to the user during his selection. This text box is not for user's input, so it should be blurred whenever it acquires focus. Whenever the user's input in any of the above three text boxes changes, if the input is not valid, this text box should display "NaN", and if the input is valid, the total cost of the user's order is recalculated and displayed in this text box. This calculation is to be done completely at the client side using JavaScript.
    d. A collection of three radio buttons that are labelled as Visa, MasterCard and Discover. This is for the user to input his payment method.
    e. A Submit button. On completing the form, the user clicks this button to submit his order to the web server.
2. Write a server-side PHP program that receives the user's order when the Submit button on the above HTML document is clicked. On receiving the order, the server-side PHP program computes the total cost of the user's order and returns an HTML document to the user as a receipt. The receipt should specify the user's name, what are ordered and the payment method in the form of a table. In addition, the server-side PHP program must also update a file named "order.txt" stored on the web server to reflect the new order. The file records the total numbers of apples, oranges and bananas ordered by all users so far in the following format:
    Total number of apples: 12
    Total number of oranges: 23
    Total number of bananas: 35

1. HTML Document

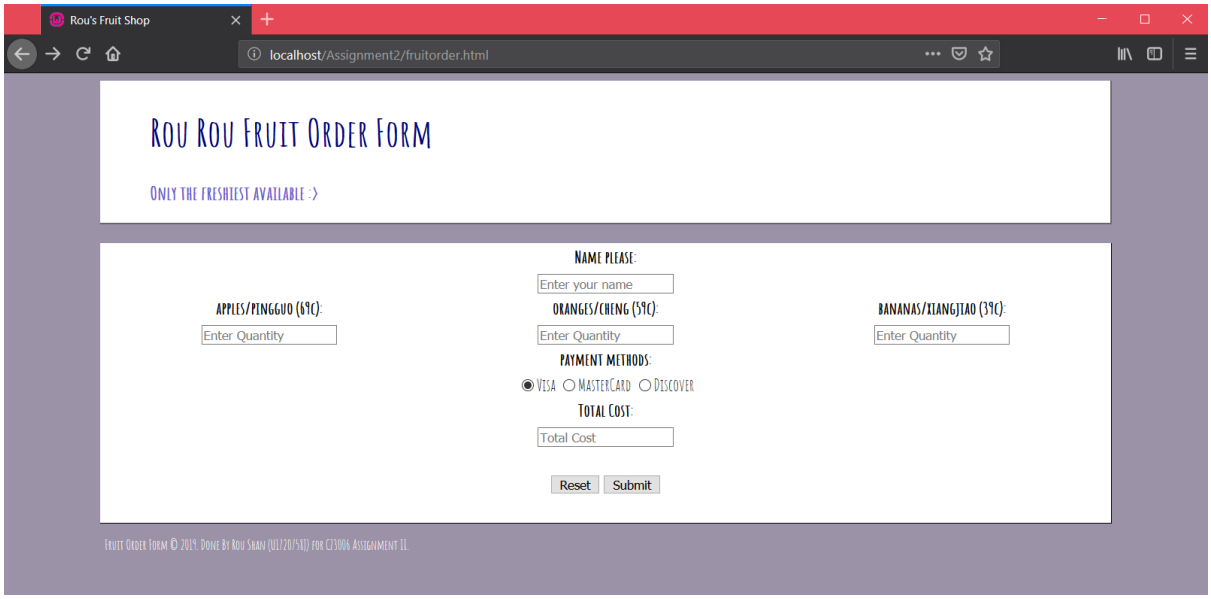

Figure 1 Screenshot of website's user interface.

1.1. User's Name Input

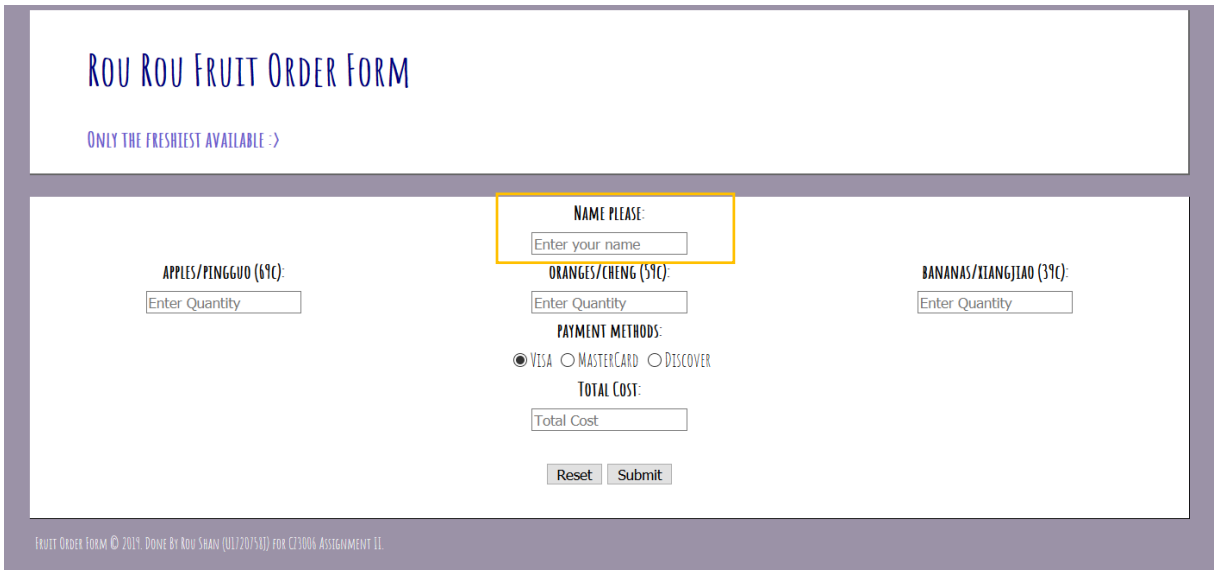

Figure 1.1.1 User's name input enclosed in box

User would have to enter their name in the text box under "Name Please:" as shown in Figure 1.1. If this input field is being left empty, a warning message will appear to prompt the user to enter their name before submitting the form.

Figure 1.1.2 Screenshot when username field is left empty

```
22          <tr>
23              <th colspan="3">Name please:</th>
24          </tr>
25          <tr>
26              <td colspan="3"><input type="text" name="cusName" placeholder="Enter your name" required=""></td>
27          </tr>
```

Figure 1.1.3 Code implementation of a required text field for user's name.

1.2. Apples/Oranges/Bananas Order

In this segment, validation is done to prevent the user from entering invalid input into the quantity text boxes on apple, orange and banana. As the quantity text box is mainly meant for entering numeric characters, users are not allowed to enter other input such as alphabets and special characters. If the user has entered a non-numeric character, it will prompt an alert message informing the user on the invalid input in the respective fruit's quantity textbox and request user to re-enter that field with the correct input. All these validations are done in JavaScript with the script tag in HTML. The code implemented is shown on figure 2.1.

```
71  <script type="text/javaScript">
72      //Check if input is valid or not. If invalid, the text box will display "NaN" and issue an error message.
73      function validateNum(){
74          numApple = document.forms["orderForm"]["apples"].value;
75          numOrange = document.forms["orderForm"]["oranges"].value;
76          numBanana = document.forms["orderForm"]["bananas"].value;
77
78          //Vaildate input for order apple textbox which can only contain numbers
79          if(isNaN(numApple)){
80              alert("Invalid input at apples! Please enter a valid integer. :)");
81              return false;
82          }
83
84          //Vaildate input for order orange textbox which can only contain numbers
85          else if(isNaN(numOrange)){
86              alert("Invalid input at oranges! Please enter a valid integer. :)");
87              return false;
88          }
89
90          //Vaildate input for order banana textbox which can only contain numbers
91          else if(isNaN(numBanana)){
92              alert("Invalid input at bananas! Please enter a valid integer. :)");
93              return false;
94          }
95
96          else{
97              //Calculate the total cost of all fruits if all inputs are valid
98              return true;
99          }
100     }
101
```

Figure 1.2.1 Code implementation for validation of fruit's quantity in respective textboxes.

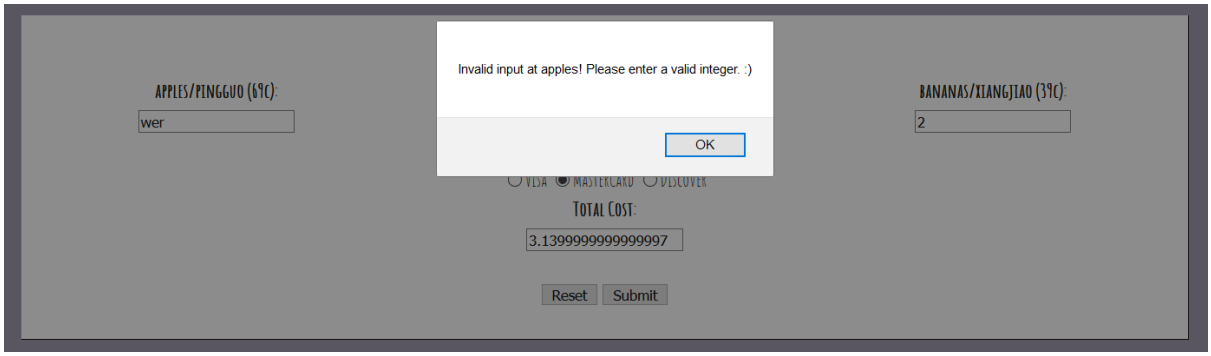

Figure 1.2.2 Screenshot of alert message when invalid input is being entered

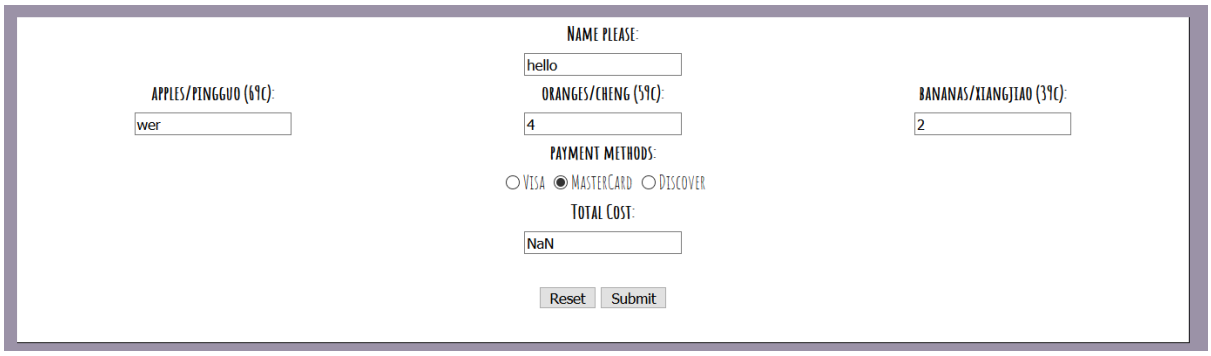

Figure 1.2.3 Screenshot after user clicks “OK” button from the previous page (Figure 2.2)

As shown in Figure 2.2 and 2.3, when an invalid input is entered, the document will do a validation on the inputs and display an alert message accordingly. Thereafter, the total cost will be displayed as “NaN” at the total cost textbox.

1.3. Total Cost

The total cost textbox shows the total cost of the order based on the quantity of fruits the user has input. The document will retrieve the values from the fruits’ textboxes after validation is done and do the appropriate calculation. The result from this calculation will then be displayed in the total cost textbox. These are done in JavaScript with the script tag in HTML.

```
102     //If input is valid, the total cost of the user's order is recalculated and displayed in totalCost textbox
103     function calculateCost(){
104         numApple = document.forms["orderForm"]["apples"].value;
105         numOrange = document.forms["orderForm"]["oranges"].value;
106         numBanana = document.forms["orderForm"]["bananas"].value;
107         totalCost = numApple * 0.69 + numOrange * 0.59 + numBanana * 0.39;
108         document.forms["orderForm"]["totalCost"].value = totalCost;
109     }
110
```

Figure 1.3.1 Code implementation for function calculateCost

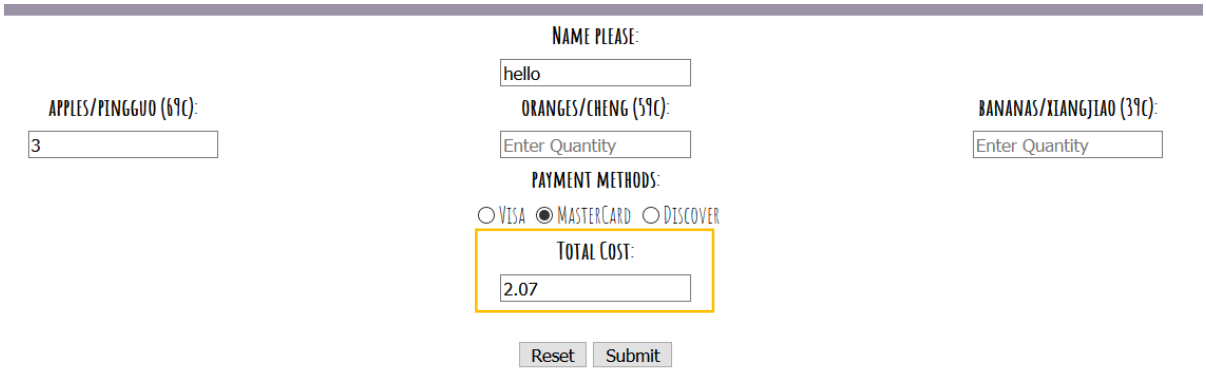

Figure 1.3.2 Screenshot after the user inputs quantity for apples

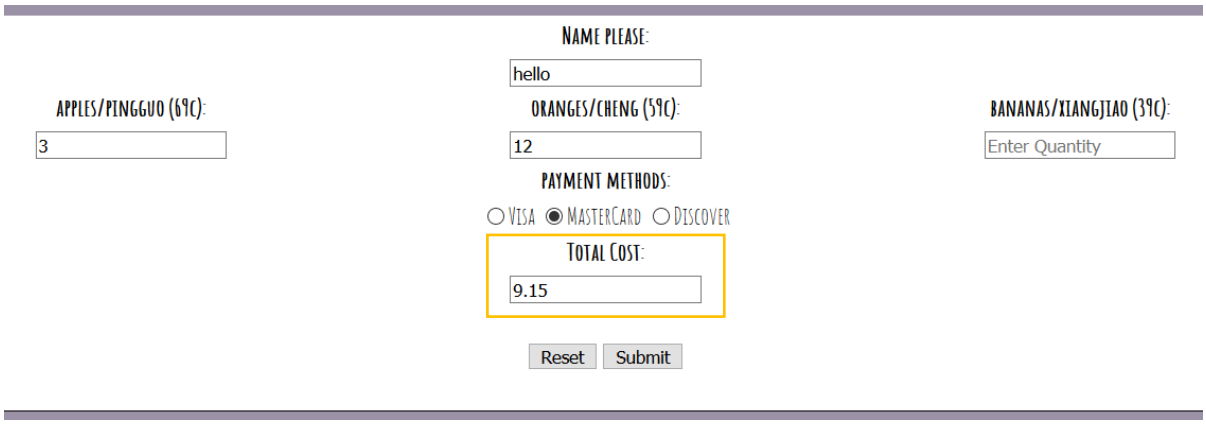

Figure 1.3.3 Screenshot after user inputs quantity for oranges

From Figures 3.2 and 3.3, the amount for the total cost changes after quantities for different fruits have been entered (highlighted in the boxes above).

1.4. Payment Method

```
38          <tr>
39              <th align="center" colspan="3">payment methods:</th>
40          </tr>
41          <tr>
42              <td colspan="3">
43                  <input type="radio" name="paymentMethod" value="Visa" checked="checked">Visa
44                  <input type="radio" name="paymentMethod" value="MasterCard">MasterCard
45                  <input type="radio" name="paymentMethod" value="Discover">Discover
46              </td>
47          </tr>
```

Figure 1.4.1 Code implementation for payment methods

This shows that the user can choose between the three payment method offered and by default, the document has already selected VISA as the default way of paying. In order to prevent user from selecting more than one radio button, once the user selects another radio button, only that particular radio button will be checked. This is implemented by declaring the same name tag (paymentMethod) for all three radio buttons.

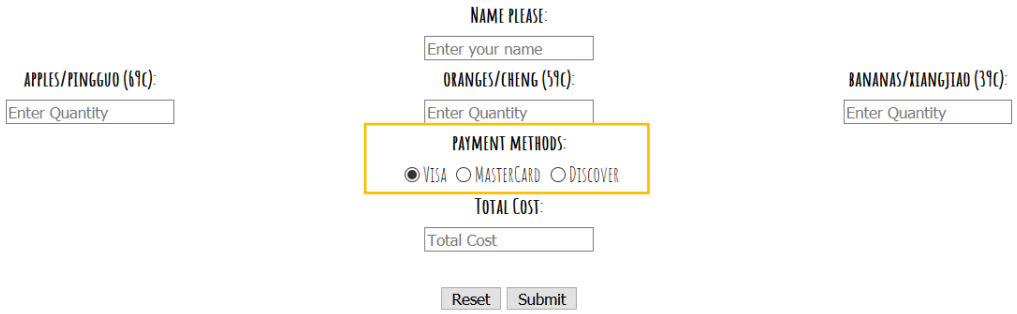

Figure 1.4.2 Screenshot of the default payment method

1.5. Submit Order

```
54          <tr>
55              <td colspan="3"><br>
56                  <input type="reset" value="Reset">
57                  <input type="submit" value="Submit"></td>
58          </tr>
```

Figure 1.5.1 Code implementation for reset and submit buttons

2. PHP Document

Once the user has clicked on the submit button and the inputs have all been validated, the html document will pass all the input values via PHP server to “submit.php” file. This .php document will communicate with the local PHP server. Once it’s able to retrieve all the input values, it will display the grand total cost for the user. Once done, it will generate an order receipt to the user in the form of an HTML document. The code implementation is shown in figure 2.1.

```php
<!-- Backend PHP Code -->
<?php

//Retrieving of form inputs
$customer = $_POST["cusName"];
$apple = $_POST["apples"];
$orange = $_POST["oranges"];
$banana = $_POST["bananas"];
$payment = $_POST["paymentMethod"];
$totalCost = $_POST["totalCost"];

//Opens order.txt or create a new order.txt if it does not exist
$myfile = fopen("order.txt", "w") or die("Unable to open file!");

//Update quantity of fruits
$txt = "Total number of apples : " . $apple;
fwrite($myfile, $txt);
$txt = "\r\nTotal number of oranges : " . $orange;
fwrite($myfile, $txt);
$txt = "\r\nTotal number of bananas : " . $banana;
fwrite($myfile, $txt);

//Store cotents to order.txt
fclose($myfile);
?>

```

Figure 2.1 Code implementation for backend PHP code

For each variable declared, “$_POST” will retrieve the variables passed from the fruitOrder.html to the current script via the HTTP POST method which in this case.

```
    <title>Order Confirmation :-)</title>
</head>
<body>
    <table align="left" width="30%" bgcolor="#FFFFFF">
        <tr>
            <td><h1>Receipt for <?php echo $customer; ?></h1></td>
        </tr>
        <tr>
            <th>Name: </th>
            <td> <?php echo $customer; ?></td>
        </tr>
        <tr>
            <th>No. of apples: </th>
            <td> <?php echo $apple; ?></td>
        </tr>
        <tr>
            <th>No. of oranges: </th>
            <td> <?php echo $orange; ?></td>
        </tr>
        <tr>
            <th>No. of bananas: </th>
            <td> <?php echo $banana; ?></td>
        </tr>
        <tr>
            <th>Payment by: </th>
            <td> <?php echo $payment; ?></td>
        </tr>
        <tr>
            <th>Total Cost: </th>
            <td> $ <?php echo $totalCost; ?></td>
        </tr>
    </table>

</body>
```

Figure 2.2 Code implementation for order receipt in HTML

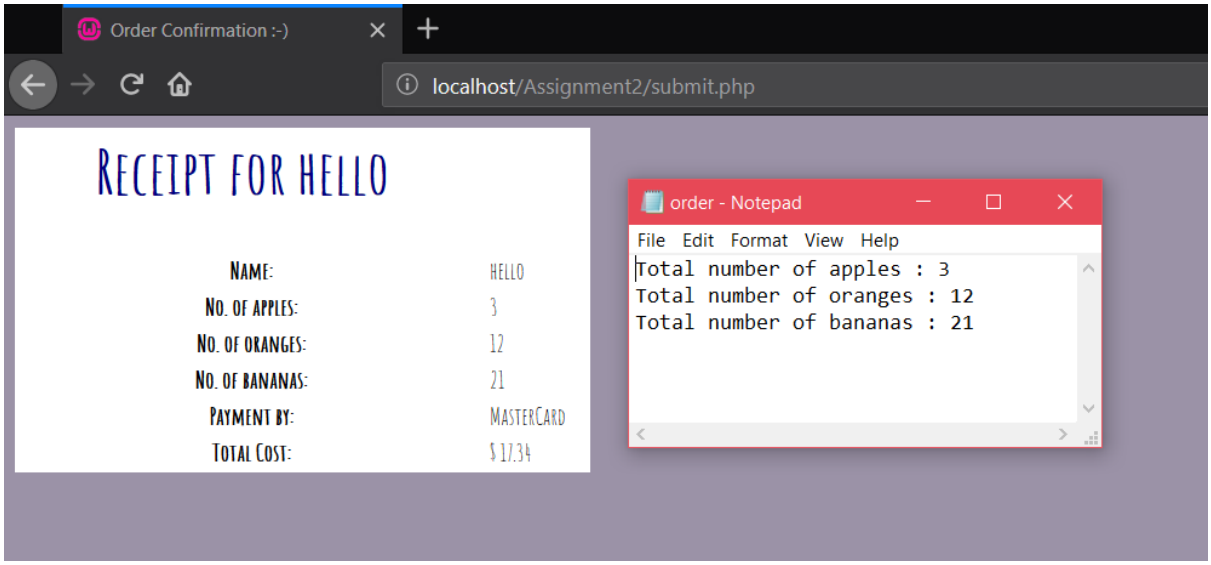

Figure 2.3 Order Receipt for user in HTML and Notepad

3. Source Code

fruitOrder.html

```
<!DOCTYPE html>
<html>
<!------ Main Index HTML Fruit Order Form ---------->
    <head>
        <!-- CSS -->
        <link rel="stylesheet" type="text/css" href="style.css">
        <link href="https://fonts.googleapis.com/css?family=Amatic+SC" rel="stylesheet">
        <title>Rou's Fruit Shop</title>
    </head>

    <body>

    <div style="text-align:center;">
        <table align="center" bgcolor="#FFFFFF" width="85%" style="border-right:2px solid #666666;
            border-bottom:2px solid #666666;">
            <tr><td align="left">
                <h1>Rou Rou Fruit Order Form</h1>
                <h2>Only the freshiest available :> </h2>
            </td></tr>
        </table>

        <br>
        <form name = "orderForm" action = "submit.php" method = "post">
            <table id="t01" align="center" bgcolor="#FFFFFF" width="85%">
                <tr>
                    <th colspan="3">Name please:</th>
                </tr>
                <tr>
                    <td colspan="3"><input type="text" name="cusName" placeholder="Enter your name"
                        required=""></td>
                </tr>
                <tr>
                    <th>apples/pingguo (69c):<br></th>
                    <th>oranges/cheng (59c):<br></th>
                    <th>bananas/xiangjiao (39c):<br></th>
                </tr>
                <tr>
                    <td><input type="text" name="apples" placeholder="Enter Quantity" onblur="validateNum();
                        calculateCost(); "><br></td>
                    <td><input type="text" name="oranges" placeholder="Enter Quantity" onblur=" validateNum();
                        calculateCost();"><br></td>
                    <td><input type="text" name="bananas" placeholder="Enter Quantity" onblur=" validateNum();
                        calculateCost();"><br></td>
                </tr>
                <tr>
                    <th align="center" colspan="3">payment methods:</th>
                </tr>
                <tr>
                    <td colspan="3">
                        <input type="radio" name="paymentMethod" value="Visa" checked="checked">Visa
                        <input type="radio" name="paymentMethod" value="MasterCard">MasterCard
                        <input type="radio" name="paymentMethod" value="Discover">Discover
                    </td>
                </tr>
                <tr>
                    <th colspan="3">Total Cost:</th>
                </tr>
                <tr>
                    <td colspan="3"><input type="text" name="totalCost" placeholder="Total Cost" onfocus="this
                        .blur()"></td>
                </tr>
                <tr>
                    <td colspan="3"><br>
                        <input type="reset" value="Reset">
                        <input type="submit" value="Submit"></td>
                </tr>
            </table>
        </form>
    </div>
    </body>

    <footer>
        <p class="text-uppercase">
           Fruit Order Form &copy; 2019. Done By Rou Shan (U1720758J) for CZ3006 Assignment II.
        </p>
    </footer>
```

```
    <script type="text/javaScript">
        //Check if input is valid or not. If invalid, the text box will display “NaN” and issue an error message.
        function validateNum(){
            numApple = document.forms["orderForm"]["apples"].value;
            numOrange = document.forms["orderForm"]["oranges"].value;
            numBanana = document.forms["orderForm"]["bananas"].value;

            //Vaildate input for order apple textbox which can only contain numbers
            if(isNaN(numApple)){
                alert("Invalid input at apples! Please enter a valid integer. :)");
                return false;
            }

            //Vaildate input for order orange textbox which can only contain numbers
            else if(isNaN(numOrange)){
                alert("Invalid input at oranges! Please enter a valid integer. :)");
                return false;
            }

            //Vaildate input for order banana textbox which can only contain numbers
            else if(isNaN(numBanana)){
                alert("Invalid input at bananas! Please enter a valid integer. :)");
                return false;
            }

            else{
                //Calculate the total cost of all fruits if all inputs are valid
                return true;
            }
        }//end validateNum()

        //If input is valid, the total cost of the user’s order is recalculated and displayed in totalCost textbox
        function calculateCost(){
            numApple = document.forms["orderForm"]["apples"].value;
            numOrange = document.forms["orderForm"]["oranges"].value;
            numBanana = document.forms["orderForm"]["bananas"].value;
            totalCost = numApple * 0.69 + numOrange * 0.59 + numBanana * 0.39;
            document.forms["orderForm"]["totalCost"].value = totalCost;
        }//end calculateCost()

    </script>

</html>
```

style.css

```css
/*-- CSS --*/
body {
  background-color: #9B92A7;
  font-family: "Amatic SC", cursive;
}

h1 {
      font-size: 40px;
      color: navy;
      margin-left: 50px;
}

h2 {
    font-size: 20px;
    color: slateblue;
    margin-left: 50px;
}

p {
    font-size: 14px;
    color: white;
    margin-left: 100px;
}

/*-- orderForm table --*/
table#t01 {
    font-size: 18px;
    border-right: 1px solid black;
    border-bottom: 1px solid black;
}
```

submit.php

```php
<!-- Backend PHP Code -->
<?php

    //Retrieving of form inputs
    $customer = $_POST["cusName"];
    $apple = $_POST["apples"];
    $orange = $_POST["oranges"];
    $banana = $_POST["bananas"];
    $payment = $_POST["paymentMethod"];
    $totalCost = $_POST["totalCost"];

    //Opens order.txt or create a new order.txt if it does not exist
    $myfile = fopen("order.txt", "w") or die("Unable to open file!");

    //Update quantity of fruits
    $txt = "Total number of apples : " . $apple;
    fwrite($myfile, $txt);
    $txt = "\r\nTotal number of oranges : " . $orange;
    fwrite($myfile, $txt);
    $txt = "\r\nTotal number of bananas : " . $banana;
    fwrite($myfile, $txt);

    //Store cotents to order.txt
    fclose($myfile);
?>

<!-- Frontend HTML Code -->
<html>

    <!-- CSS -->
    <link rel="stylesheet" type="text/css" href="style.css">
    <link href="https://fonts.googleapis.com/css?family=Amatic+SC" rel="stylesheet">

    <head>
        <title>Order Confirmation :-)</title>
    </head>

    <body>
        <table align="left" width="30%" bgcolor="#FFFFFF">
            <tr>
                <td><h1>Receipt for <?php echo $customer; ?></h1></td>
            </tr>
            <!-- Name of Customer -->
            <tr>
                <th>Name: </th>
                <td> <?php echo $customer; ?></td>
            </tr>

            <!-- Fruits Ordered -->
            <tr>
                <th>No. of apples: </th>
                <td> <?php echo $apple; ?></td>
            </tr>
            <tr>
                <th>No. of oranges: </th>
                <td> <?php echo $orange; ?></td>
            </tr>
            <tr>
                <th>No. of bananas: </th>
                <td> <?php echo $banana; ?></td>
            </tr>

            <!-- Payment Method -->
            <tr>
                <th>Payment by: </th>
                <td> <?php echo $payment; ?></td>
            </tr>

            <!-- Grand Total Cost -->
            <tr>
                <th>Total Cost: </th>
                <td> $ <?php echo $totalCost; ?></td>
            </tr>
        </table>

    </body>
</html>
```