
Lab report

CZ30006 Net centric computing assignment 2

Name: Xavier Tan

Matriculation number: U1720697B

Lab group: TS6

File organization:

Index.html : This html file has all the hyper text markup language document and it will reference to a bootstrap frame work library for css(cascading style sheet) as well as, additional internal css script which is written under the style tag. This script is mainly for alignment of the document objects in the html file.

In the css folder, bootstrap.min.css: This is the minified(no space) bootstrap cascading style sheet. I am using this open source front-end component library for the aesthetics of the web page.

In javascript folder, validator.js: This file will include all the code needed to do the client sided input validation as well as calculating the total cost. Further explanation below.

orderForm.php: This is php file whereby it will print the receipt and create or write to the order.txt file

order.txt: A text file consisting of the accumulated orders.

photos folder contains the respective photos used in the webpage

Implementation:

A) HTML & CSS :

The script will follow the tree structure of DOM(document object model) i.e. roughly depicted as:

```
<head>
  <title> </title>
</head>
<body>
  <table>
    <tr>
      <th> </th>
      <td> </td>
      <td> </td>
    </tr>
    <tr>
      </tr>
  </table>
</body>
```

At the start of the document it will reference to the 2 external files, bootstrap.min.css and validator.js. The input objects will later call the functions in validator.js when it detects a change from the user to validate the user's input.

Following that is the internal css denoted with the style tag, as mentioned earlier, this is mainly to alignment the elements so that it will be centred and the width size is controlled. Percentage and auto margin is used such that the webpage is compatible with multiple screen sizes.

Following that will be the head element where title name is set, navigation bar and its sub elements defined. Although the navigation bar has no much purpose now, it gives a better feel to the website, and can be extended later on. For example, the "home" element will hyper link back to the home page which this page.

After that will be the body element. The form element will exist here. The form element will encapsulate all of the input elements in body section. As the information in the body is needed to be passed to the server-side later on. The first input tag is used for user to input their name. It has attribute of type = "text" hence any text can be given into the input, name and id attribute to be used for referencing from javascript and php. Placeholder attribute to tell user what to do in that box and lastly a required property to make it compulsory field. Tables are used to structure the layout of the webpage. tr tag for rows, th tag for header and td for cells. A table will have a head and body. In my table's head it will consist the headers of the table, which is namely, Fruits, Cost/Each and Amount. By default the placeholder for the amount input is 0, this is just a translucent number that gives the user some example what to input that the text box. When the user interacts with the webpage, for example typing an input to any of the input box under amount of fruits to be bought. The attribute on change in input tag will be activated. For example, onchange="changeApple(this.value)" will pass the value that the user input to javascript for input validation. In other words, when the user interacts with the webpage, html will sense that an event has occurred and trigger the javascript function in response. This will happen for other fruits also. Tag id and name attributes is given for the input tags so that it can be referenced later on from javascript and php. For the total cost text box, same thing happens, when the user focuses on the text box, it will sense that an event has happened, and then in response it will use the blur() function which is inbuilt in html5 to disallow input from user even though the input tag has attribute of type text. All input tags have required property for first layer of check, to make sure that user must enter

something in the compulsory text fields. Data validation and more error checking is done in the javascript file.

Within each row of fruits, the first cell from the left will include a small picture of the fruit. This is done by using the img tag with src equals to the directory of the photo which i saved locally. Inline css is also used to align the photos.

For the payment methods, radio input type is used for the circular check box. They are all the same name so that at most one radio can be checked, but id is different so that later when printing receipts PHP knows which payment method is used. Required is also used so that the user must choose a payment type to pay for the fruits. img tag is also used like previously to display photos stored locally.

For submit button, the button has type = "submit". So when the button is clicked it will send the data to the server side via POST method as mentioned earlier. A listener is used again, to make sure that all input is valid. This is done through the onclick function in the button tag, so when html senses a click from user, it will activate the isEmpty() function in javascript to check all the data inputs before sending the input to server.

The css is mainly referenced to the file using class = "name of the style" in the tags. Then the predefined style in bootstrap file will be applied to my html file. There are also a few inline css used for the small images in the text box. These are mainly used to define the placement of the images, at a corner.

B) JavaScript

From javascript, there are mainly two ways to access the document objects in HTML file. document.getElementById("ID") and document.getElementsByName("Name"). The former is used mostly in my javascript. The totalCost() function will do the calculations to get the total cost for all the fruits multiplied by its respective cost. Since this is money it is fixed to 2 decimal places.

Then the total value is assigned to document.getElementById("____ID____"), hence this function can also be used to manipulate the elements. Each fruits have their own validation function so that the webpage will not notify the user when other fruits have not been given any input, and that the error alerts will be given at the correct time. In the

`changeApple(input)/changeOrange(input)/changeBanana(input)`, the input value of what the user typed in the text box is passed to the javascript function. In this function, it checks for correct type of input, making sure the input is a number through the `isNaN` function. The `isNaN()` function determines whether a value is Not a Number. This function returns true if the value equates to NaN. Otherwise it returns false. So if it is not a number input type is wrong. Negative input check is simple done by checking if it is lesser than 0. To check if the input is an integer, just need to parse the number using the function `parseInt(input)`. This will change any number that the user input and round it to the nearest integer if needed. Then just check if the original data is equal to this parsed data. If it is different, it means the number input has been rounded off meaning the data input is not an integer.

If it fails any of these checks, the submit button will disabled and an alert will notify the user's error type. It will automatically delete the input given by user so that they will be forced to change the input to correct type. Submit button is disabled so that when the user wants to submit, they will be able to, hence being forced to check for any input error. Else, if all the checks passes, the submit button will be enabled and the `totalCost()` function will be called.

In the `totalCost()` function, it calculates the total cost of all the fruits bought by the user. Here a further check is done to make sure that the cost is more than 0, meaning there is no erroneous data input and that at least 1 fruit is bought by the user. This check is done by using logical operator `&` to make sure that not all amount bought = 0. If the user buys no fruits, an alert will notify the user and submit button will be disabled until the user buys at least 1 fruit from 1 type of fruit. If there was any wrong user data input Total will show "NaN", as required by the lab assignment.

A payment type must also be selected to submit. This is done though both html required tag and javascript.

When the user clicks submit button, the function `isEmpty()` checks all fields again to make sure all fields are filled properly. The fields include the input, and method payment type. If the user did not fill in the correct type of data, the field would be empty straight away so, this two mechanism forces the user to try to type the correct data, only then the user can submit. Multiple ways of error checking is introduced to ensure, that in most cases at least one of this checks work, for example the user might not be using HTML5, so checks will be done by javascript, or if javascript is disabled by the user, HTML will at least do some basic validation.

C) Server Side, php

In the server side, a php script is written. At the start there is check to see if all the input given using the isset function, this an added measure to prevent any errors in case the browser disables javascript. After that the variables are initiated by the form values passed by html via the POST method. i.e \$name = \$_POST["nameInput"]; Whereby variable name will get the value from the input box with attribute name = "nameInput".

Variable \$numberRegex = "/\d+/"; /d/ is a shorthand character class, which matches all numbers for 0 to 9. This is used for matching later in preg_replace. Variable \$fileName = "order.txt"; defines the text file to be written on.

To write into a text file. Firstly, if this file does not exist, it will create the file using the file_put_contents function, and write for the string defined in the variable \$contents with the corresponding values gotten using \$_POST method from HTML. Else if it already exist, it will open the file in read mode, initialise an empty string variable which is to be appended to later. A for loop is used, from the start to the end of file (feof function). Within this loop fgets(\$file) function is used to get the string data in the corresponding line. The first line is for apple, second line for orange and third for banana. preg_match(\$numberRegex, \$string, \$matches); The reg_match is used to perform a regular expression match between integers (0-9) using the variable numberRegex defined earlier and the entire line. The number matched will be the fruitsAmount since there is only 1 number in 1 line. Hence the matched index = 0. Then using preg_replace, the number is replaced by the new total amount given by \$matches[0] + \$apple/orange/bananaAmount. This content is then appended to the empty string variable initialised, \$content, earlier on. This process is repeated for 3 time, each time for 1 line(1 line for 1 type of fruit). Then all the content is written to the file using file_put_contents function. fclose() is used to close the file, as good practise and to reduce chances of data lost.

After that a receipt is printed through the echo function, which echoes out the html document for the receipt. The receipt uses a table to show the data. Within this html script that is going to be echo out, it will also contain the values from the html input page via the POST method. This is done by putting variables as the text.

Source code below if needed, excluding bootstrap

HTML and internal CSS

```
<!DOCTYPE html>
```

```
<html>
```

```
<script type = "text/javascript" src = "javascript/validator.js"></script> <!--
```

```
Include javascript-->
```

```
<link href="css/bootstrap.min.css" rel="stylesheet"> <!--bootstrap,
```

```
relationship is stylesheet-->
```

```
<!--Style to center the body and table-->
```

```
<style>
```

```
.center {  
  margin-left: auto;  
  margin-right: auto;  
  width: 80%;  
}  
body {  
  text-align: center;  
  background-repeat: repeat-x;  
}
```

```
h1 {  
  text-align: center;  
  margin: auto;  
  color: white;  
}
```

```
</style>
```

```
<!--The head of html, consisting of the navigation bar-->
```

```
<head>
```

```
<title >Fruit Stall</title>
```

```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-  
target="#navbarColor01" aria-controls="navbarColor01" aria-expanded="false"  
aria-label="Toggle navigation">
```

```

    <span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id = "navbarColor01">
    <span><h1> FruitsBar </h1> </span>
    <ul class="navbar-nav mr-auto">
        <li >
            <a class="nav-link" href="index.html">Home</a>  <!-- go back to front
page-->
        </li>
        <li>
            <a class = "nav-link">about us </a>
        </li>
    </ul>
</div>
</nav>
</head>
<!--Entire body is the form, will pass whatever is in the form to php using post
method , table is the fruits and text input area-->
<body>
    <div class = "center">
        <form action = "orderForm.php" id = "orderForm" method = "post" name =
"orderForm">
            <br>
            Your name please:<input type = "text" name = "nameInput" placeholder =
"Input your name here" id = "nameInput" required></input>
            <div id = "food table" class = "center">
                <br><h4>Fruit Selection</h4>
                <table border = 3 class = "center, table table-hover" >
                    <thead>
                        <tr>
                            <th scope="col">Fruits</th>
                            <th scope="col">Cost/Each</th>
                            <th scope="col">Amount</th>
                        </tr>
                    </thead>
                    <tr>
                        <td>Apple</td>

```

```

        <td>$0.69</td>
        <td> <!-- for is for form-->
        <input type = "text" name = "AppleAmount" id = "AppleAmount"
placeholder = "0" onchange="changeApple(this.value)" required/></br>
        </td> <!-- id is for js/html , name is for php-->
    </tr>
    <tr>
        <td>Orange</td>
        <td>$0.59</td>
        <td>
        <input type = "text" name = "OrangeAmount" id = "OrangeAmount"
placeholder = "0" onchange="changeOrange(this.value)" required/></br>
        </td>
    </tr>
    <!--3rd row-->
    <tr>
        <td>Banana</td>
        <td>$0.39</td>
        <td>
        <input type = "text" name = "BananaAmount" id = "BananaAmount"
placeholder = "0" onchange="changeBanana(this.value)" required/></br>
        </td>
    </tr>

    <tr>
        <td>Total Cost</td>
        <td></td> <!--blank space-->
        <td>
        <input type = "text" name = "Total" id = "Total" onfocus="blur()"
placeholder="0" > </br>
        </td>
    </tr>
</table>
</div>
<!--Payment method-->
<div class = "center">
    <h6> Select Payment method </h6>

```

```

<table border = 2 class = "center" style = "width: 50%">
  <tr>
    <td>Visa</td> <!-- margin for picture top and down 0,
left and right 10-->
    <td><input type = "radio" name = "radio-btn" value="Visa" id = "Visa"
required></td>
  </tr>
  <tr>
    <td>MasterCard</td>
    <td><input type = "radio" name = "radio-btn" value = "MasterCard" id =
"MasterCard" required></td>
  </tr>
  <tr>
    <td>Discover</td>
    <td><input type = "radio" name = "radio-btn" value = "Discover" id =
"Discover" required></td>
  </tr>
</table>
</div>
<!--Submit button-->
<!--checks for incomplete or improperly answered fields.-->
</br>
<button type = "submit" name = "submit" class="btn btn-warning"
style="color: black" id = "submitbutton" onclick = "isEmpty()">Submit</button>
</br>
</br>

</form>
</div>

</body></html>

```

JavaScript

```
function totalCost(){
  // calculations
  var appleCost = parseInt(document.getElementById("AppleAmount").value)*0.69;
  var orangeCost = parseInt(document.getElementById("OrangeAmount").value)*0.59;
  var bananaCost = parseInt(document.getElementById("BananaAmount").value)*0.39;
  var total = appleCost + orangeCost + bananaCost;
  // Money is 2 decimal
  total = total.toFixed(2);
  // make sure that calculation is done correctly should not have anything negative or less
  than 0
  if(total > 0){
    document.getElementById("Total").value = "$"+total; // change value in total's text box
    to the value
    document.getElementById('submitbutton').disabled = false; // enables the submit
    button
  }else if(appleCost == 0 && orangeCost == 0 && bananaCost == 0 ){ // user typed all 0.
    alert("Buy at least 1 fruit!");
    document.getElementById('submitbutton').disabled = true; // disables the submit
    button
  }else{ // all other wrong data input
    document.getElementById("Total").value = NaN;
    document.getElementById('submitbutton').disabled = true; // disables the submit
    button
  }
}
//////////////////input validation functions//////////////////
// alert function will notify user's error
// Each fruits have their own validation function so that the webpage will not notify the
user when other fruits have not been given any input.
function changeApple(input){
  // parseInt returns an integer rounded down. Chooses the first Integer in the text, if there
  is any spaces
  var inputInt = parseInt(input);
  // Making sure that only numbers is given
  if (isNaN(input)){
    document.getElementById('submitbutton').disabled = true; // disables the submit
    button
    document.getElementById("Total").value = NaN; // Change the total value
```

```

    alert("Amount input is invalid, use integers only!");
}
else if (input < 0){ //negative input
    document.getElementById('submitbutton').disabled = true; // disables the submit
button
    document.getElementById("Total").value = NaN;
    alert("Wrong amount, more than 1 please");
    document.getElementById('AppleAmount').value = null; // erases the input given by
user so that they will be forced to change the input to correct type
}
else if(inputInt !== input || inputInt == NaN ){
    document.getElementById('submitbutton').disabled = true; // disables the submit
button
    document.getElementById("Total").value = NaN; // Change the total value
    alert("Integer value only");
    document.getElementById('AppleAmount').value = null; // erases the input given by
user so that they will be forced to change the input to correct type
}
else{
    document.getElementById('submitbutton').disabled = false; // enables the submit
button
    totalCost();
}
}

function changeOrange(input){
    // parseInt returns an integer rounded down. Chooses the first Integer in the text, if there
is any spaces
    var inputInt = parseInt(input);
    // Making sure that only numbers is given
    if (isNaN(input)){
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Amount input is invalid, use integers only!");
    }
    else if (input < 0){ //negative input
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Wrong amount, more than 1 please");
        document.getElementById('OrangeAmount').value = null; // erases the input given by
user so that they will be forced to change the input to correct type
    }
}

```

```

    else if(inputInt !== input || inputInt === NaN ){ // a decimal number given such that
whats given by user is different from the parsed number, or no input given
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Integer value only");
        document.getElementById('OrangeAmount').value = null; // disables the submit
button
    }
    else{
        document.getElementById('submitbutton').disabled = false; // enables the submit
button
        totalCost();
    }
}

function changeBanana(input){
    // parseInt returns an integer rounded down. Chooses the first Integer in the text, if there
is any spaces
    var inputInt = parseInt(input);
    // Making sure that only numbers is given
    if (isNaN(input)){
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Amount input is invalid, use integers only!");
        totalCost();
    }
    else if (input < 0){ // negative input
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Wrong amount, more than 1 please");
        document.getElementById('BananaAmount').value = null; // erases the input given by
user so that they will be forced to change the input to correct type
    }
    else if(inputInt !== input || inputInt === NaN ){ // a decimal number given such that
whats given by user is different from the parsed number, or no input given
        document.getElementById('submitbutton').disabled = true; // disables the submit
button
        document.getElementById("Total").value = NaN; // Change the total value
        alert("Integer value only");
        document.getElementById('BananaAmount').value = null; // erases the input given by
user so that they will be forced to change the input to correct type

```

```

    }
    else{
        document.getElementById('submitButton').disabled = false; // enables the submit
button
        totalCost();
    }
}

function isEmpty() { // checks if all fields are complete
    if (document.getElementById("AppleAmount").value == "" ||
document.getElementById("OrangeAmount").value == "" ||
document.getElementById("BananaAmount").value == "" ||
document.getElementById("nameInput").value == "" ||
(document.getElementById("Visa").checked == false &&
document.getElementById("MasterCard").checked == false &&
document.getElementById("Discover").checked == false))
    {
        alert("Please complete all fields.");
        return false;
    }
}

```

OrderForm

```

<?php
    if(isset($_POST["nameInput"]) && isset($_POST["AppleAmount"]) &&
isset($_POST["BananaAmount"])
    && isset($_POST["OrangeAmount"]) && isset($_POST["Total"]) && isset($_POST["radio-
btn"])) {
        // variable initialisation, POST method is used to pass from html to php
        $name = $_POST["nameInput"];
        $numberRegex = "/\d+/" ; // a shorthand character class, which matches all
numbers
        $appleAmount = intval($_POST["AppleAmount"]);
        $orangeAmount = intval($_POST["OrangeAmount"]);
        $bananaAmount = intval($_POST["BananaAmount"]);

        $total = $_POST["Total"];

        $paymentMethod = $_POST["radio-btn"];

        $fileName = "order.txt";

```

```

        if (!file_exists($fileName)) {
            // This code block expected to run only once
            $contents = "Total number of apples: ".$appleAmount."\r\nTotal
number of oranges: ".$orangeAmount."\r\nTotal number of bananas: ".
$bananaAmount."\r\n";
            file_put_contents($fileName, $contents); //create file and put the
contents into the file
        }
        else {
            $file = fopen($fileName, "r");
            $contents = "";
            for ($i = 0; !feof($file); ++$i) {
                $string = fgets($file);
                preg_match($numberRegex, $string, $matches); //reg_match
— Perform a regular expression match
                switch($i) {
                    case 0: // .= Concatenation assignment
                        $contents .= preg_replace($numberRegex,
$matches[0] + $appleAmount, $string); //preg_replace — Perform a regular expression
search and replace
                        break;
                    case 1:
                        $contents .= preg_replace($numberRegex,
$matches[0] + $orangeAmount, $string);
                        break;
                    case 2:
                        $contents .= preg_replace($numberRegex,
$matches[0] + $bananaAmount, $string);
                        break;
                }
            }
            fclose($file); //close the file, clear cache
            file_put_contents($fileName, $contents); //the existing file is
overwritten
        }
        //print out receipt
        echo
        '<html>
<style>
.center {
margin-left: auto;
margin-right: auto;
width: 80%;

```
