Subject: Stage 2 Digital Technologies

Task: Assessment task 4/ Web store

Student: George Gadalla

SACE NO: 591604J

A picture containing drawing

Description automatically generated

Teacher: Mr Papazoglou

Table of Contents

[1. Introduction 3](#_Toc43480377)

[2. Part A (final product) 3](#_Toc43480378)

[3](#_Toc43480379)

[2.1 Mark-up design 4](#_Toc43480380)

[4](#_Toc43480381)

[3. Part B (code analysis) 5](#_Toc43480382)

[3.1 Description outcome 5](#_Toc43480383)

[3.2 Total order discussion 6](#_Toc43480384)

[3.3 Innovative features 7](#_Toc43480385)

[3.4 Effectiveness vs Efficiency 9](#_Toc43480386)

[3.5 Improvements to making code more effective/efficient 10](#_Toc43480387)

[4. Further evaluation 10](#_Toc43480388)

[5. Improvements and Conclusion 10](#_Toc43480389)

# Introduction

**The scenario**: I am web application development specialist who has been assigned to create a clothing store. In the initial phase, this web portal allows customers to match shirts and pants only.

This will be completed through the use of Komodo edit 9 and CSS stylesheets, HTML and JavaScript. The final objective is to have the HTML and CSS be complementary to my screen which is the average size for a laptop and have it visually appealing and functional to the user. The main outcome is the user will be able to use the website to calculate the costs of the clothes.

# Part A (final product)

**Logo:** The logo is in the top right corner with the brand name clear

# A picture containing person, man, standing, woman Description automatically generated

**Background image:** the image shows a man and a women wearing clothing that promotes the brand (the image is called 2.png

**Button:** the button says “get started” and is animated to quickly transition from blue to white when the mouse is over

**Title:** The name of the brand is centred and clear to the user

# 

The background image is in a div called ‘intro’ that contains everything, the h1 tag is the name of the company ‘Joker clothing’. The joker logo is in the logo div which is within the logoholder div. The button was placed in a class called to btn and was then edited to have its own hover effects.

This was a minor page, is severed its purpose by promoting brand in a simple way with the man and the women wearing clothes from the business. It only had one button that redirected the user to the main page.

**Table:** The table below is where all the JavaScript is outputted, the **dList** dropdown list is an array outputted from the JavaScript.

**nbar:** This div contains the images of the head and the h2 tag

A picture containing monitor, wooden

Description automatically generated

**div:** Each tile is named accordingly to with the name ‘div’ followed by the number

**Radio button:** All the radio buttons are referenced to the main module “ething();” including the dropdown list.

**Logo:** The logo is centred similar to the last websites and the brand name is clear

## 2.1 Mark-up design

**Header:** The header contains the company logo and is simple with a black colour

A picture containing photo, sitting, green, room

Description automatically generated

**Colour scheme:** The colour scheme is simple blend of green, black and white (\*note it could change)

**Images:** The images shows the user what product they will be buying if they choose the option

My design will include 2 separate pages, when the user presses on the link they will be firstly be redirected to a landing page with a button asking them if they would like to shop. Accompanied by a background image of some clothes and the logo somewhere in the corners. If the user presses the button on the landing page they will be sent into the second page with the images of the pants and shirts with radio buttons for the user to choose if they want to buy pants and shirts or only one item. There will also be a dropdown menu that will ask the user for what type of shipping they would like and will display the price of each option

Using JavaScript modules can be created for cases depending on what the user chooses to buy. There will be multiple fragments of JavaScript code that will do other actions, I want to have the images of the clothes and the radio button on one ‘tile’ appear through on scroll animations the page will be large in height to have the clothes spaced apart with the final pay option at the bottom of the page. The last part of the code using JavaScript will be creating the sub module to calculate the total price (including $40 administration fee) that the user will have to pay.

# Part B (code analysis)

## 3.1 Description outcome

The main objective will give the user the option to select some shirts and pants or only one option and shipping costs and then calculate the total cost for the user. This will be done with JavaScript through getting values of radio buttons and referring to them into an array to be used as variables in the code. This will further be modified using an ‘if’ statement to correctly calculate the cost of the options without any complications. Then in order to run the full code correctly it will placed in a main module to run the code each sub-module at a time to ensure correct calculations.

## 3.2 Total order discussion

**Code 1 (Total order calculation)**

First the module that activates all the code from here

function ething(){

appear1();

calculateTotal();

totaloptions();

appear1(); tells the table to appear through jQuery in a sub module

totalshipping();

}

This submodule shows how the total cost of everything is calculated including the admin fee

function calculateTotal(){

var adminfee = 40;

var total = adminfee + getShirtprice() + getPantprice() + getshipprice();

var tnow = document.getElementById('totalPrice');

tnow.innerHTML = total;

}

var shirt\_prices = new Array();

This is an example of how an array has been used for the radio button values and then calculates the option selected

shirt\_prices["Shirt1"]=100;

shirt\_prices["Shirt2"]=150;

shirt\_prices["Shirt3"]=180;

shirt\_prices["Noshirt"]=0;

function getShirtprice(){

var Shirtprice=0;

var theForm = document.forms["Options"];

var selectedshirt = theForm.elements["shirtSelected"];

for(var i = 0; i < selectedshirt.length; i++)

This is an if statement that has been used to determine the value of the shirt selected

{

if(selectedshirt[i].checked)

{

Shirtprice = shirt\_prices[selectedshirt[i].value];

break;

This is where the shirt’s price is outputted

}

}

return Shirtprice;

\*NOTE this method was repeated for calculating the pants and the shipping

This code (**Code 1**) was planned prior to experimenting with the coding, writing different methods. For this to logically work, I planned to create a main module and with that module some sub-modules, this was the simplest and most effective and efficient method to creating the solution. The idea is that this main module is referenced to the radio buttons and the dropdown list, so the code was immediately show a change in the table to avoid having to press on the checkout button every time to make a calculation.

This module was created and was referenced In the code in both the radio buttons and the dropdown list as follows.

**Code 2 (output of main module)**

<input type="radio" name="shirtSelected" value="Shirt1" onclick="ething();">Shirt 1 ($100)<br>

<select name="dList" id= "dList" style="width:275px; background: linear-gradient(to right, #a0d800, #00ff8f); height: 35px; border: solid black; cursor:s-resize;" onchange="ething()">

As seen in **Code 2** the highlighted section(s) shows how when the button/dropdown list is clicked each time it initiates ething and executes the code in order shown in **structure chart 1.**

**Structure chart 1**

This outputs the full cost of the purchase in the total section of the table

calculateTotal

totalshipping

ething

First the module allows the ‘proceed to checkout’ button to appear which makes the table with costs appear

Appear1

totaloptions

This is the last submodule that is outputted, it shows the total shipping cost

This it shows the total options of clothes cost only rather than including the shipping

## 3.3 Innovative features

**AOS**

AOS (animation on scroll):

AOS is a JavaScript built function that is sourced from the internet that triggers an animation when scrolling down, this function can also be applied to appear on arrival to the website. AOS has been used for a lot of features of this website, including everything in the header div and the company logo. All the tiles that contain the items like the shirts and pants have been animated to appear on approval as-well.

AOS has been used to make the website generally feel more professional and look more appealing to the user.

**Code 3 (HTML)**

<div class="div" **data-aos**="fade-left" **data-aos-duration**="800" **data-aos-once="true"**>

<fieldset style="width:275px; position: absolute; border: none; ">

<img src="images/ws.png"><br>

<input type="radio" id="shirt" name="type" value="1">

<label for="male">Shirt 1 ($100)</label><br>

</fieldset>

</div>

The part of **Code 3** that is highlighted in yellow is where the AOS is declared to work within the div, the **data-aos** determines the type of animation that will occur, AOS has a library of different animations with different names for different effects. The **data-aos-duration** determines the amount of time in milliseconds (800 = 0.8 seconds) that the animation will take to load fully into the screen. This part **data-aos-once="true"** determines how many times the animation will load into the page, however by default it does load in once, the installation guide recommended to have that statement written to avoid any issues with use.

**jQuery**

jQuery is similar to AOS, it’s an externally sourced and can be used for editing the CSS within the JavaScript in functions. jQuery has been specially used for the animations in this code.

**Code 4** (JavaScript)

function appear(){

$('#table').fadeIn(400);

$('.info').fadeOut(400);

}

**Code 4**, shows how the table was able to fade into the screen on click of the button, this was done using the section highlighted in yellow that shows that the div called table was set to ‘**fadeIn’** an animation feature of jQuery. Through this application the table was able to fade in, keeping in consideration that the div was to be hidden prior to clicking the button. The ‘(**400**)’ determines the milliseconds it will take for the table to appear. The **‘fadeOut’** function in jQuery was the opposite of the fade in, the info div disappeared for the table div to appear.

**Shortcut icon**

**Code 5**

<link rel="shortcut icon" type="image/png" href="images/si.png" ><!this sources the image in tab>

The shortcut icon is a feature of HTML that allows designers to attach an image to the tab to make the website stand out. Looking at **Code 5** the image is outputted in a **link** tag and is referenced as a ‘**shortcut icon**’.

**Iframe**

**Code 6**

<iframe src="https://maps.google.com/maps?q=Skid%20Row%2C%20Los%20Angeles&t=&z=13&ie=UTF8&iwloc=&output=embed"></iframe><br>

Iframes can be used to reference an external webpage into an existing one. Through **Code 6** it can be seen that Google maps address has been linked using the **src** (source) feature. There were some complications to this as it had to be converted and generated first to fit the div.

## 3.4 Effectiveness vs Efficiency

**Effective but inefficient: AOS**, it is visual appealing to any user that lands on the website, however integrating it into the code could have been better and done another way, although the code does work correctly it’s hard to read.

**Ineffective and inefficient:** The method the **dropdown** list was **styling**, It could have styled in a more convenient manner to avoid some complication in the code. All the CSS was done in the HTML select tag. This resulted in lines of CSS that ran down the HTML document. This could have been placed into a div instead to make it more simple.

**Efficient and ineffective:** The **Radio button integration** was efficient as it did work well in the final result, users are able to use the buttons to select what shirt they want to purchase without any issues. The main issues came in the JavaScript were referring to their button value’s, it could have been done in a more effective manner.

**Effective and efficient:** The way the code works and the how it is able to **calculate the costs** correctly and shipping without any issues. It does accomplish the purpose of the website and does work in best and fastest way. It contains a reasonable level of complexity through the application of the modularisation in JavaScript and using arrays to refer to form value to collect the information needed to output in the JavaScript.

## 3.5 Improvements to making code more effective/efficient

**AOS:** Possible improvements to making the AOS more efficient, would be creating a ‘sub-div’ for all the tiles animated from the right of the screen instead of having it done twice it would only be coded once for that type of animation to avoid redundant/repetitive code.

**Dropdown list:** The dropdown list could have been styled in a div in the CSS file rather than within in the select tag to avoid styling in the HMTL.

**Radio buttons:** The radio buttons in the JavaScript could have coded better and more conveniently to calculate the costs, some possible improvements could be using more appropriate button ID’s to make referring to them easier in the JavaScript, another possibility could be using another method other than modularisation.

# Further evaluation

Overall the website did work and all the Innovative features were installed correctly and added some uniqueness to the website, it make the page look overall more professional and more enjoyable to be used. The AOS was the main features that stood out as every div was animated to appear on load with a different animation for most div.’s. Minor things that could be reconsidered are visual changes such as the selected colour scheme and the shape of the tiles, and the time it takes for them to load In (in milliseconds). However a major thing that could have been improved would be positing of the footer to fit the screen rather than having to scroll down to see it. A minor change could have been making the image size(s) larger to be viewed on smaller devices.

# Improvements and Conclusion

In summary, the website was a success and worked visually well and in functionally. If this was re-created the major differences would be in the way the code is structured, limiting the number of div’s to make styling easier and within less lines of code. Further differences would include the colour-scheme and the size of the images and the tiles to make the website more appealing than it currently is to the user. Hence this was an generally a well-designed website in functionality as the JavaScript was successfully created to make the correct calculations without major complications, and the appearance was also visually impressive to view and easy to look at.