



# Edge Hill University

Department of Computer Science

**CIS2152**  
**Fundamentals of Web Coding**  
**Level 5**

Coursework  
2019/2020

**Module Leader:** Quanbin Sun

Callum Gill – 24040461

## Contents

<b>Introduction</b>	<b>3</b>
<b>Instructions</b>	<b>3</b>
<b>Features</b>	<b>3</b>
<b>Test Document</b>	<b>7</b>
<b>Conclusion</b>	<b>12</b>

# Introduction

In this report I will be making a report on the vending machine I created. Firstly I will show instructions on how to operate the vending machine. Secondly I will show the code of my features and say what the code does. and lastly I will show my testing document i did while creating the code.

## Instructions

1. The program will ask you to enter credits to the program.
2. Enter credits into the program.
3. You will be asked to choose from 4 options these being Products, View Credits, Top-up and Refund.
4. (Products) If products were chosen, then the products will be shown to the user to pick from.
5. Then pick a number from the list of products.
6. When the product is picked it will then show you what you have purchased and then ask you if you want to quit the program or not.
7. Then the user can repeat if they want.
8. (View Credits) If view credits were chosen, then the credits will be shown to the user as the option is selected.
9. Then the main menu will appear again.
10. (Top-Up) If top-up were chosen then a prompt will appear asking the user to enter an amount they would like to top-up.
11. Then the main menu will appear again.
12. (Refund) If refund was chosen, the program will refund the user and then show a log that they have been refunded and the user now has zero balance.
13. Then the main menu will appear again.

## Features

### View the list of products (each product must have a name & price)

The view products feature allow the user to be able to see what all the products and prices of the product is before purchase so they can know how much each product is and then know if they can afford the product or not with how much credits they have entered previously. Firstly after the user has added credits into the system to begin with the user will then come across their first menu with four options one of these options being view products. Once this option is chosen an array list will appear in the log showing the user the different products and their prices and below is a picture of how the code is implemented into my program.

```
//This is the sub menu from the main menu
function menuSelection(selection) {
  //products page
  if (selection == 0){
    products_menu([' Dr Pepper 75', ' Coke 75', ' Pepsi 75' ,
      ' Flake 150', ' KitKat 150', ' Daim 150', ' Skips 100',
      ' Monster Munch 100', ' McCoys 100', ' Refund']);
  }
  //Allow the user to see their credits
```

## View the amount of credit the user has

The view credits feature allow the user to see how much credits the user currently has in their balance this feature then will allow the user the user to quickly check how much credits they have so they can see how much products they can buy without just mindless buying products till an error message appears telling them they don't have enough balance. Also it helps the user keep track of their credit after purchase or top-up allowing them to see their credit and not have to add or subtract in their head. Firstly after the user has added credits into the system to begin with the user will then come across their first menu with four options one of these options being view credits if the user chooses view credits option then text should come up in the log telling the user their remaining credits also after purchase or top up it will also tell the user the balance after every transaction to keep the user updated. Below in the first picture is the code implementation how the user views their credits in the main menu after the user chooses the view credits option in the main menu. The second picture shows the code Implementation of the user being updated about their balance after purchasing a product.

### Main menu view credits

```
78 //Allows the user to see thier credits
79 if (selection == 1){
80
81     console.log("Your remaining credits are: " + balance) ;
82
83 }
84 //refind allow the user to update their credits
```

Allowing the user to see updated balance after purchase

```
//Updating credits after purchase
function showCredits(){
    //Updating for cans and checking if balance is correct
    if (choice == 0 && balance >= 75 || choice == 1 && balance >= 75 || choice == 2 && balance >= 75){
        balance = balance - cans;
        console.log("Your remaining credits are: " + balance);
    }
    //updating for chocolate and checking if balance is correct
    if (choice == 3 && balance >= 150 || choice == 4 && balance >= 150 || choice == 5 && balance >= 150){
        balance = balance - chocolate;
        console.log("Your remaining credits are: " + balance);
    }
    //updating for crisps and checking if balance is correct
    if (choice == 6 && balance >= 100 || choice == 7 && balance >= 100 || choice == 8 && balance >= 100){
        balance = balance - crisps;
        console.log("Your remaining credits are: " + balance);
    } else {
        console.log("You dont have enough credits please top-up in the main menu");
    }
}
```

## Add credit

The add credits feature allows the user to add credits into the program so they can buy products later on in the program. There is also a second way to add credits into the program with top-up function this will again be located in the main menu as top-up. This will allow the user to keep adding credits into the program without having to quit the program to add credits again. When using the top-up function it should add the newly added credits with the old credits creating a new balance so the user can then keep buying products if they chose so buy adding new credits if they run out of credits from what they originally added into the system. When the program first loads it will ask the user to enter credits, so then the user will then add how many credits they would like to add into the program this balance will then be saved throughout the whole program. Then to add to add more credits the user has to select the top-up option in the main menu once this has happened the user will get prompted by a question asking them to add credits the user will then add credits of their choice and the program will then add the old balance with the new topped up balance to create a new balance for the user. The first picture is the implemented code for how the user get asked to add credits into the program at the start of the

program as you can see I use a while loop this is to check the user adds an amount above 75 credits as this is the lowest value product in the vending machine so there would be no point entering a value lower than this. The while loop works by setting the while loop to false and every time the user enters a value below 75 the if statement will run saying cannot accept that amount if the amount is above or equal to 75 the code then goes to the else statement which will then change the while loop to true ending the cycle and allowing the user to carry on with the program. The second picture is the implemented code for how the user will top-up for more credits during the program. To add credits it uses the function setCredits() so extra code is not needed to be written as it serves the same purpose the old balance will be saved and then after the new balance is set they will be added together to create a new balance.

### Adding credits in the beginning

```
//setting credits for the user
function setCredits(){

    var test = false;

    while(test==false){
        console.log('Please Enter your credits');
        //Asking the user to add credits into the system
        var balance = readline.question('Enter credits here: ');
        console.log('Your credit balance is: ' + balance);
        //Due to the lowest cost item being 75 this will stop the user putting anything below 75 into the program
        if(balance < 75){
            console.log('Cannot accept anything below 75');
        }else{
            test = true;
        }

        return balance;
    }
}
```

### Topping up credits during the program

```
//top-up allows the user to top up
if (selection == 3){
    //saving the old balance
    oldBalance = balance;
    balance = setCredits();
    //adds the old balance the user had with the new balance the user is adding into the program
    balance = parseInt(balance) + parseInt(oldBalance);
    console.log("Balance Now: " + balance);
}
```

### Purchase a product

The purchase a product features allows the user to purchase products from the view products selection it will also check to see if the user has enough credit to purchase the product as it wouldn't make sense on a business point of view if user could purchase products if they didn't have enough credits. When the user purchases a product it will also update the balance by minusing the product price with the current balance. When the user selects the view products option in the main menu it will ask the user for a choice this is the option the user will use to buy a product. So the user will then make a choice for this example, the user will choose option 1 then it will show the user what they have purchased for example it will sat you have purchased coke and also show the user their new balance after purchase so in this example minus 75 credits of whatever balance the user had. In the first picture is the implemented code for showing the user what product they have purchased, the if statements works from the choice from 1-8 and also make sure their balance is enough and then the log will then correspond with whatever choice the user made. The second picture shows the code implementation of the product being purchased it checks to see what choice the user made and then also check if the has enough credits if true it will minus the product from the balance and then also show the user their remaining balance if none are true it will then tell the user they don't have enough credits and the user has to top-up in the main menu.

Telling the user what they have purchased

```
//This function is from the array to allow the user to see what they have purchased
function purchase(choice){
  if (choice == 0 && balance >= 75){
    console.log("You have purchased Dr Pepper");
  }
  if (choice == 1 && balance >= 75){
    console.log("You have purchased Coke");
  }
  if (choice == 2 && balance >= 75){
    console.log("You have purchased Pepsi");
  }
  if (choice == 3 && balance >= 150){
    console.log("You have purchased Flake");
  }
  if (choice == 4 && balance >= 150){
    console.log("You have purchased KitKat");
  }
  if (choice == 5 && balance >= 150){
    console.log("You have purchased Daim");
  }
  if (choice == 6 && balance >= 100){
    console.log("You have purchased Skips");
  }
  if (choice == 7 && balance >= 100){
    console.log("You have purchased Monster Munch");
  }
  if (choice == 8 && balance >= 100){
    console.log("You have purchased McCoys");
  }
}
```

Updating after purchase

```
}
//Updating credits after purchase
function showCredits(){
  //Updating for cans and checking if balance is correct
  if (choice == 0 && balance >= 75 || choice == 1 && balance >= 75 || choice == 2 && balance >= 75){
    balance = balance - cans;
    console.log("Your remaining credits are: " + balance);
  }
  //Updating for chocolate and checking if balance is correct
  if (choice == 3 && balance >= 150 || choice == 4 && balance >= 150 || choice == 5 && balance >= 150){
    balance = balance - chocolate;
    console.log("Your remaining credits are: " + balance);
  }
  //Updating for crisps and checking if balance is correct
  if (choice == 6 && balance >= 100 || choice == 7 && balance >= 100 || choice == 8 && balance >= 100){
    balance = balance - crisps;
    console.log("Your remaining credits are: " + balance);
  } else {
    console.log("You dont have enough credits please top-up in the main menu");
  }
}
```

## Ask for a refund of their credit

The refund feature allows the user to refund their credit this means if a user is done with the program and what to leave they can refund their credits and use them enough day instead of them being wasted and lost in the program. Firstly after the user has selected the refund option in the main menu it will then refund the credits then show a log saying your credits have been refunded then set the balance to 0 to make it look like the credits have been refunded then a new log will be shown to the user saying balance now with the updated balance after the refund has happened. The picture below is the implemented code for showing the refund method firstly it will show the log to the user saying the credits have been refunded. Then it will set the balance to 0 and then show the user the new balance so they can see they have been refunded.

```
//refund allows the user to refund their credits
if (selection == 2){
    console.log("You have been refunded your credits");
    balance = 0;
    console.log("Balance Now: " + balance);
}
```

## Test Document

### Testing add credit feature

Test Example	Test Description	Test Expected Outcome	Test Actual Outcome	Action Taken
Testing if the text appears in the log asking for the user to add credit to the program	The text in the log asking the user to add credit to the program	The text will appear in the log and the user will be able to see it	The text appeared in the log and the user was able to see it	N/A
Testing if the balance updates	When the user enters credit the balance updates	The balance will update to what the user has entered	The balance updated to what the user has entered	N/A
Testing what happens when the user tries to add less than 75 credit	User adds less than 75 credits to the program	The program will return text to the user asking for them to re-enter a new amount as 75 is too little of an amount to enter please enter again	The expected outcome worked, and the user was asked to re-enter the amount into the program	N/A
Testing what happens when the user tries to add more than 75 credit	User adds more than 75 credits to the program	The program will accept the amount and allow the user to move on with the program	The expected outcome worked, and the user could move on with the program	N/A

When the user did add an amount less than 75 does the program then allow the user to re-type an amount so they can continue	New question appears after the user typed an amount less than 75 asking them to re-type the amount of credits, they want to enter	The program will ask the user to re-enter the amount of credits they want to enter into the program	At first this didn't work too well as the program just ended after the user went to put an input in again	The action that was taken was to add a while loop to allow the code to keep running through the program
Testing if the top-up page appears after the user selects the top-up option	From seeing the main menu, the user selects the view page and then the top-up page will appear	The top-up options will appear after the user has selected the top-up option	The top-up option appeared after it was selected	N/A
Testing when the user chooses the option top-up that the user will be asked how much they would like to top-up	A log should appear after the user has clicked on the top-up menu option	A prompt should appear asking the user how much credits do they want to top-up	The expected outcome came true and the log appeared asking the user how much credits do they want to top-up	N/A
Testing when the user tries to add credit via the top-up function and the old credit balance adds on top of the newly added credit that has just been added	The user has entered a number into the top-up feature and this number should then add onto the previous balance	After the user has entered the number they want to top-up by the new balance should appear telling them what their new balance is now	The expected outcome came true and after the user entered a number to top-up by the previous credit was added onto the new top-up credit	N/A

### Testing view credit feature

Test Example	Test Description	Test Expected Outcome	Test Actual Outcome	Action Taken
Testing if the main menu appears after the user adds credits with one of the options being view credit	The four options in the main menu will appear for the user to choose from one being view credit	The four options view products, view credit, top-up and refund	The four options appear in the main menu for the user to choose from one being view credit	N/A
Testing if the view credits option appears after the user selects the view credits option	From seeing the main menu, the user selects the view credits page and then the view credit option will appear	The view credits options will appear after the user has selected the product option	The view credits option appeared after it was selected	N/A



Testing if the user can then carry on with the program after checking how much credits they have	After the user sees how much credits they have can they carry on with the program	The user will see their credits and then want to carry on with the program a question will appear asking them if they want to carry on if they do they will just go back to the main menu	The expected outcome came true and the user was able to carry on with the program after viewing credits	N/A
Testing if the view balance/credit feature works and shows the user their credit after the program runs once and balance is updated via purchase or top-up	In the main menu there is the view credit option when this is chosen the user will be able to see their balance	When the view credit option is chosen the user will get to see what credit/balance they currently have after	The expected outcome came true and the user was able to see their current credit/balance and was correctly updated after a purchase or top-up	N/A

### Testing refund feature

Test Example	Test Description	Test Expected Outcome	Test Actual Outcome	Action Taken
Testing if the main menu appears after the user adds credits with the four options appear one of them being refund option	The four options in the main menu will appear for the user to choose from one of these options being refund option	The four options view products, view balance, top-up and refund	The four options appear and one being refund in the main menu for the user to choose from	N/A
Testing if the refund page appears after the user selects the refund option	From seeing the main menu, the user selects the refund option and then the refund option will appear	The refund options will appear after the user has selected the refund option	The refund option appeared after it was selected	N/A
Testing if the refund feature works and resets the balance to zero and tells the user the refund has been complete	In the main menu there are four options one of them being refund and when the user clicks on this option it should refund all the money back to the user and then tell the user the money has	After the refund option has been chosen the balance of the user should be zero and the program should then come up with a log telling the user they have	The expected outcome came true and the refund worked and when the user clicked on the option their credits were refunded and a log came up telling them they have	N/A

	been refunded and the users balance is now zero	been refunded and their balance is now zero	been refunded and the balance is now set to zero	
--	---	---	--	--

### Testing view list of products feature

Test Example	Test Description	Test Expected Outcome	Test Actual Outcome	Action Taken
Testing if the main menu appears after the user adds credits with the four options appear one of these options being view products	The four options in the main menu will appear for the user to choose from one being view products	The four options view products, view balance, top-up and refund	The four options appear in the main menu one being view products for the user to choose from	N/A
Testing if the view products option appears after the user selects the product option	From seeing the main menu, the user selects the view product option and then the product page will appear	The product options will appear after the user has selected the view product option	The view product option appeared after it was selected	N/A
Testing once the view product page has been chosen the list of products will then appear with their prices also	After the user has selected the view products option in the main menu the list of products with their prices should appear for the user to see	The product list should appear with the name of the product and prices in a list for the user to see	The expected outcome came true and products were presented to the user in a list with the name of the product and the price of the product	N/A

### Testing purchase a product

Test Example	Test Description	Test Expected Outcome	Test Actual Outcome	Action Taken
Testing once the view product page has been chosen the list of products will then appear with their prices also	After the user has selected the view products option in the main menu the list of products with their prices should appear for the user to see	The product list should appear with the name of the product and prices in a list for the user to see	The expected outcome came true and products were presented to the user in a list with the name of the product and the price of the product	N/A
Testing once the products have been listed the user	After the user has seen the products and their prices a	The expected outcome is that a question should	The expected outcome came true and the question	N/A

should then be prompted by a question	question should appear asking them what product they would like to purchase	appear asking the user what product they would like to purchase and the user be able to respond and chose a product	was asked to the user and they were able to reply to the question and chose a product	
Testing on the product option when the user selects an option the correct log will correspond with the selection.	if the user picks coke the log will appear with you have purchased coke etc.	The correct option will appear for this instance it will say coke has been purchased	The expected outcome did not happen as no matter the option it always returned the first option in the array	This code was updated with the choice option needed to be called in the product function and adding this in the function allowed for the code to get past the first array option
Testing if the balance is lower than the product	If the user has less credit that the cost of a product then the program will tell the user they don't have enough credit and go back to the main menu to top-up	The user has 95 credits to buy a chocolate bar but the product cost 100 so the program should say you dont have enough credit please top-up in the main menu	The expected outcome came true and the user was told they didn't have enough credit and needed to top-up in the main menu	N/A
Testing if the balance updates after purchase with enough credits	When the user selects a product will the correct amount be taken of the balance	Whatever option the user picks for this example skips the balance taken of the user should be 150 so 150 should be subtracted of the user	The expected outcome did come true and the balance was updated to whatever option the user has selected	N/A
Testing if after the user has purchased a product can they keep using the program if chosen to or leave the program	After the user has purchased a product the user will then be prompted a question asking them if they want to continue or not	Depending if what the user selects the program will end or continue. If the user selects 'Y' or 'y' the code will end if anything else is imputed the program will restart	The expected outcome did come true and if the user selected 'Y' or 'y' and program ends if the user selects anything else the program will restart	N/A

## Conclusion

In review I feel like this project of creating a vending machine went well I managed to complete all the features which were Product list, View Credit, Add Credit, Purchase product and ask for a refund. In the report I have then go on to talk about what each feature does and how it was implemented into the code, the instructions on how to run the code and the testing documents.