Web Development – Mr. Goldman

Level 2 Project – Fast Food Cashier

**Project Overview**

You are a cashier at a fast food restaurant. Throughout the course of your shift, customers will come in and order items off the menu. As the cashier, it's your job to take the orders, take the money from the customer, and give the proper change. At the end of the day, the computer will determine whether or not your register is complete.

**The Site**

The user's register begins with $100 in the following denominations:

* 100 pennies
* 40 nickels
* 50 dimes
* 40 quarters
* 42 one dollar bills
* 4 five dollar bills
* 2 ten dollar bills

The register should be depicted graphically, with a button for each denomination. The buttons will be used for giving change.

At the start of the program, the user is beginning a 2 hour shift. The program will continue to run until the 2 hour shift is complete.

As soon as the shift begins, a customer will come up to the register and place an order. It will take 2 to 7 minutes (determined randomly) to serve that customer. After the customer has ordered and paid, there will be a 0 to 2 minute break until the next customer arrives.

There should be a “clock” on screen which shows how much time is left in the user’s shift.

A customer will make an order based on the following guidelines:

The customer will order 0 or 1 random entrée.

The customer will order 0, 1, or 2 random side dishes.

The customer will order 0 or 1 random dessert.

The customer will order 0 or 1 random drink.

The entrees are:

* Hambuger $1.99
* Chicken Sandwich $1.99
* Veggie Sandwich $2.29

The sides are:

* French Fries $.99
* Salad $1.39
* Cheese Sticks $1.49
* Rice $1.19

The desserts are:

* Ice Cream $1.89
* Pie $1.69
* Cookie $.89

The drinks are:

* Soda $1.19
* Bottled Water $1.25
* Juice $1.69

Once the order has been made, the total will be calculated and the customer must pay. There are two options for paying and they will be determined randomly. The customer will either give exact change (meaning that the program must figure out the denominations and add them to the register) or the customer will use the bills in his/her wallet. When paying in this manner, the customer will use the following method in determining payment:

The customer will have 0 - 10 one dollar bills.

The customer will have 0 - 5 five dollar bills.

The customer will have 0 - 3 ten dollar bills.

The customer will have 0 - 1 twenty dollar bill.

The customer will first attempt to pay with 1 dollar bills. Once (s)he is out of ones, (s)he'll move onto fives, tens, and the possible twenty. Keep a running total in the background as the user pays with these bills. When there is enough to pay for the meal, the page will display the total amount of cash that the customer has given (not the denominations) and the amount of change owed the customer. It will also add the money to cash register.

The user must now give change by clicking on the buttons on the drawer. Once the user has given the proper change (or more if there is not enough of a certain denomination), the break and the next customer will immediately follow.

At the end of the shift, the computer will have a total for all of the register sales. It will add up all of the money based on the quantities of each bill and coin in the register. If the total is more than $1 short, the user will be fired. If the total is more than $10 over, the user will be fired. Anything else and the user will be asked back to work again the next day.

**Enhancements**

The user will have the opportunity to short-change the customer. Instead of automatically going onto the next customer, add a “Next Customer, Please” button. The user can click this button to dismiss the current customer after (s)he’s paid and go on to the break and the next customer.

If a customer is short-changed, there is a chance that (s)he will notice. The chance that the customer will notice is 5% + 1% for every ten cents short-changed. If the customer notices the user trying to rip him/her off, the user will be fired and the program will end.

**Programming Requirements**

|  |
| --- |
| * You must write all of your own code. Use of a drag and drop interface or the usage of code downloaded from the internet is not permitted. |
| * Use of deprecated code is not permitted. |
| * Your home page must be named *index.html*. |
| * You must comment your name into the top of every page, but below the doctype. |
| * Your code must be structured in a consistent and legible manner. |
| * Your interface must be smooth and easy to figure out for a client. |

*You project is to be submitted via Moodle. Compress all of the necessary files into a zip file.*