

Tsarbucks

Summary

Using PHP, HTML, and a connection to a MySQL database, create an e-commerce system for a fictional coffee company. This involves a shopping cart for a customer, a menu of items that can be added to the cart, and a separate screen for a barista to mark ordered items as complete.

Make sure you have an understanding of form submissions, mysqli or PDO, sessions, strings, and arrays.

Check out the screenshots for an idea of how the system works.

Specification

The Tsarbucks e-commerce system provides a way for a customer to log-in, see their order history, select drinks from a menu, add the desired drinks to their shopping cart, and submit the order to be prepared. It also provides a way for a barista to log-in and mark the items in an order as completed (i.e. when they have finished making a drink for a customer).

User Interface

The user interface should contain the following:

General

- 1.Login screen that takes a username and password
- 2.Logout functionality (link, screen, etc) that logs-out the current user
- 3.Navigation bar that has all available options for the logged-in user

Customer Component

- 1.Home screen that displays information (this can be the same as the My Orders screen)
- 2.My Orders screen that displays the details of past and current orders
 - 1.If an item in an order has been completed, it should show that it has been completed. Otherwise it should show as pending.
- 3.Menu screen that gives the customer a set of items to add to the cart
- 4.My Cart screen that displays everything currently in the customer's cart
 - 1.Each item on the screen should have the option to remove it
 - 2.There should also be a button that allows the customer to submit their order

Barista Component

- 1.Home screen that displays information (this can be the same as the Pending Orders screen)

2. Pending Orders screen that displays all orders that are currently pending

1. All non-completed (pending) items in all orders should be shown
2. Each item should have the option for the barista to mark it as completed

I'm not looking for anything stellar for the UI but just make sure all of the components are there.

Functionality

Redirecting with PHP

```
// redirect to the home page and then exit the script
header("Location: index.php");
exit();
```

General

1. The Login functionality needs to take a username and password, check it against the database, and then log the user in. The username and user ID should be saved to the session and then the user should be redirected to the home page.
2. The Logout functionality needs to destroy the entire session upon logout and then redirect the user back to the home page.
3. The navigation bar needs to link to all the relevant pages based on the logged-in user

Customer Component

1. The My Orders screen needs to show the details of each order both past and current from the database. The details are the following:
 1. Cost of the entire order
 2. The name of each item in the order
 3. Whether each item in the order has been completed or is still pending
2. The Menu screen needs to do the following:
 1. Display all items available for purchase
 2. Each item available needs to show its price
 3. Each item needs to have a way for the user to add it to his cart
3. The My Cart screen needs to do the following:
 1. Display all items currently in the user's cart
 2. Each item in the cart needs to show its price
 3. Each item in the cart needs to have a way for the user to remove it from his cart
 4. Provide a way for the user to submit the order in his cart
 1. This should save the data of the order to the database

Barista Component

1. The Pending Orders screen needs to do the following:
 1. Show only items that are pending in the orders (i.e. not completed) from the database
 2. Each pending item needs to show its price

3. Each pending item needs to have a way for the barista to mark it as completed
1. Marking an item as completed should update the order item in the database

Grading

This project is worth 20 points, weighted at 15% of the grade. Implementing all of the functionality is considerably more important than how the UI looks.

I'm going to be using Chrome to run everything since Firefox can be funky sometimes.

Extra Credit

The extra credit for this project involves integrating AJAX and Socket.io with the customer's My Orders screen and the barista's Pending Orders screen.

Upon the barista marking an item as complete (you'll probably want to do that with an AJAX call now) a socket message should fire to the Socket.io server (after the database update is successful) and then a message should be sent to the customer from the server in response.

When the customer receives the message (only while on the My Orders screen), the relevant order item that was just completed should change visually somehow. My implementation involves changing the background color of the table row containing the item in the order.

The extra credit is worth 2 points.