

Homework 5

Due date: Sunday June 9, 2019, 23:59pm

Submission instructions: Submit to this CrowdGrader assignment:

https://www.crowdgrader.org/crowdgrader/venues/view_venue/4363

Note: you do not have to do any reviews.

Assignment

In this assignment, you put a shopping cart and Stripe checkout on top of the shopping site you developed for Homework 4. Here are the steps.

The buy button

Start from the endpoint of Homework 4. For each product, you add an input field to write the amount, and a button to buy. One way is to do like this,

```
<div>
  <input type="number" v-model="prod.buy_amount" />
  <button v-on:click="buy(prod._idx)" class="btn red">Buy</button>
</div>
```

and a nicer way is to do as in branch

https://bitbucket.org/luca_de_alfaro/web2py_start/src/cart-stripe-singlepage/ , and precisely using this HTML:

https://bitbucket.org/luca_de_alfaro/web2py_start/src/c5a5458d3467d32175131058415efbb7240b3beb/views/default/index.html#lines-44

The shopping cart

In the branch https://bitbucket.org/luca_de_alfaro/web2py_start/src/cart-stripe-singlepage/ above, the shopping cart is implemented using localStorage; see here for example for how things are added to the cart:

https://bitbucket.org/luca_de_alfaro/web2py_start/src/c5a5458d3467d32175131058415efbb7240b3beb/static/js/default_index.js#lines-74

Instead, you have to implement a table for the shopping cart. I suggest something like:

```
db.define_table('shopping_cart',
    Field('user_email'),
    Field('product_id', 'reference product'),
    Field('amount', 'integer'),
)
```

An alternative (and perhaps simpler) implementation would be:

```
db.define_table('shopping_cart',
    Field('user_email'),
    Field('cart', 'text'), # This stores json, just like localStorage
)
```

I suggest you also implement API methods:

```
def get_cart():
    """Get the list of products in the shopping cart,
    with their quantities."""

def inc_cart():
    """Adds the specified quantity of the specified product to the cart
    (if the product is already in the cart, the quantity on order
    is incremented)."""

def clear_cart():
    """Clears the shopping cart"""
```

Add the shopping cart page

Just as done in the branch

https://bitbucket.org/luca_de_alfaro/web2py_start/src/cart-stripe-singlepage/, add a variable “page” to Vue that has value “shop” (if the user is shopping) or “cart” (if the user is viewing the shopping cart), along with a way to toggle back and forth from shopping mode to cart mode. Precisely, again like in the given branch, if the user is shopping, there should be a button called “View cart”, and if the user is viewing the shopping cart, there should be a button “Continue shopping”.

The shopping cart page should display all the products in the cart. At the bottom, it should display the cart total.

The Place Order button

At the end of the shopping cart page, put a t button. In a real site, you should use the button to implement a Stripe flow, as done in the code we saw in class. But time is limited, and debugging the interaction with Stripe may not be fun, so I will leave this as special credit for the students who want to do it. If you want me to be able to write a glowing recommendation letter, you might want to tackle this, but otherwise it's not required.

Rather, when the user presses the Place Order button, do two things.

First, send an AJAX request to the server to clear the shopping cart, as all has been bought.

Second, open a *modal* done in CSS (simply, use a vue v-if to make the div appear); look at the example in [stupid.css](#):

```
<div class="modal" id="popup1">
  <div class="twothirds rounded white padded">
    <h2>Hello from modal</h2>
    <a class="close" href="#">x</a>
    <p>bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
    <div class="right">
      <a class="close btn" href="#">close</a>
    </div>
  </div>
</div>
```

The modal should say: "Thank you for your order of \$xxx.xx" (where xxx.xx is the total), with a button to close it.

Tidy Up

Finally, check that:

- All AJAX URLs are properly signed
- No SQL injection attacks are possible
- The site looks decent