

Acceptance Test

(How our program matches our use cases)

Add/Remove Items in Cart Use Case

The test cases below show the overall code and tests the functionality of the Shopping Cart and checks to see if it matches the use case. The following samples of code tests to see if the shopping cart is successfully able to add items to the shopping cart and remove items from the shopping cart.

Add To Shopping Cart

```
it("User adds products to Shopping Cart", () => {
  for (let i = 0; i < 5; i++) {
    let index = i + 1;

    let title = `Food ${index}`;
    let details = `Description of Food ${index}`;

    let cost = index * 5;
    let image = {
      src: `./image-${index}`,
      alt: `Description of image ${index}`
    };

    let item = new Product(title, details, cost, image);
    newShoppingCart.addItem( item );
  }

  expect(newOrder.getShoppingCart().getProducts().size).toEqual(5);
});
```

Remove from Shopping Cart

```
it("Remove first item from Shopping Cart", () => {
  let item = newShoppingCart.getItem(0); // 1st item in Shopping
  Cart
  newShoppingCart.removeItem(item);
  expect(
    newOrder
      .getShoppingCart()
      .getProducts()
      .size
```

```

    ).toEqual(4);

    expect(
      newOrder
        .getShoppingCart()
        .getProducts()
        .has(item)
    ).toBeFalsy();
  });

```

Payment and Billing Use Case

Collects end-users payment and billing information and verifies if the information is valid. Sets up a demo payment card and billing details.

```

it("Proceed to Checkout and Enter Payment Info", async () => {
  newPaymentDetails.RecieveUserInformation(
    // Billing & Shipping are the same location
    "2000 Simcoe St N, Oshawa, ON L1G 0C5",
    "2000 Simcoe St N, Oshawa, ON L1G 0C5",

    // Payment Info.
    "5425233430109903 -- 123 -- 04/23 -- 1234"
  );

  expect(newPaymentDetails.getBilling()).toBe.equal("2000 Simcoe St
N, Oshawa, ON L1G 0C5");
  expect(newPaymentDetails.getShipping()).toBe.equal("2000 Simcoe
St N, Oshawa, ON L1G 0C5");

  expect(newPaymentDetails.getPaymentInformation()).toBe.equal("54252334301
09903 -- 123 -- 04/23 -- 1234");

  // We haven't signed up with a service that verifies user payment
  info.,
  // so this can't work for now, but we will still leave it in
  expect(await
newPaymentDetails.VerifyPaymentInformation()).toBe.false;
});

```

Monitor for Security and Privacy Threats

The Monitor Security and Privacy Threats test case's main purpose is to make sure that the website is secure and can not be easily hacked into. An example of how the team is doing this is by confirming payment and billing information. Once the payment and billing information is finalized by the end user, the information is verified and the information is checked to see if it is valid.

Clicking On Categories

From the end-users side of view we tested to see if the set categories were easily accessible and filtered correctly.

Accessibility Features

We utilized the browsers native features and tested to see if the following accessible features were given work while using the website and without affecting the website.