

Shopping Cart Checkout Exercise

In a typical shopping cart, you have the option to check out and then are taken through a series of steps. In this exercise, you'll work with the step that collects the shipping and billing addresses from the user.

Install Dependencies

The first thing you need to do is install any dependencies by running the command `npm install` from the command line.

Shipping and Billing variables

Next, set up a few variables to hold your form data. Add the following objects and properties:

- shipping
 - address
 - address2
 - city
 - state
 - zip
 - deliverTo
- billing
 - address
 - address2
 - city
 - state
 - zip

The fields are all strings, so you can initialize them to empty strings.

Data Binding with Input Elements

Now that you have your shipping and billing variables set up, you need to somehow bind them to their corresponding form fields. There is a way to do this in Vue. If you've forgotten how to do this, revisit the chapter in the student book.

Now that you have data binding set up with your input elements, you need to display them on the right side of the page. If you look in the markup for the `AddressInfo` component, you'll see code that looks like this:

```
<h3>Shipping Address</h3>
<div class="line-item">
  Address:
  <span class="shipping-address"></span>
</div>
<div class="line-item">
  Address 2:
  <span class="shipping-address2"></span>
```

```
</div>
<div class="line-item">
  City:
  <span class="shipping-city"></span>
</div>
<div class="line-item">
  State:
  <span class="shipping-state"></span>
</div>
<div class="line-item">
  Zip:
  <span class="shipping-zip"></span>
</div>
<div class="line-item">
  Deliver to:
  <span class="shipping-deliverto"></span>
</div>
```

You need to go through and display each of the object properties for both shipping and billing.

Tests

To consider your exercise a success, you'll need to run automated tests against it. Those tests check for the following criteria:

- When you fill out a form field, its value is displayed in the right results pane.
- When you fill out all of the shipping form fields and check the checkbox "Billing Same as Shipping":
 - All form field values are copied to billing and displayed in the right results pane.

To check your work, run the tests by running the following command from the command line:

```
npm run test:e2e
```

The command above opens the Cypress tests in a UI and can be time-consuming. If you'd rather run them all from the command prompt, use the following command:

```
npm run test:e2e-headless
```

Extra Credit

There's one more issue to fix with your application. If you enter whitespace before or after any text in one of your inputs, it causes a problem. You might not notice the problem because if you look in the right results pane, the application looks fine. If you open up the Vue Devtools though, you'll see there are spaces before and after the value. There is a way to fix this, and it was mentioned in the student book. You'll want to make this correction for every field that accepts user input.