**Use your textbook, notes, and the internet to complete the following test in 3 hours or less.**

**All code submitted must be your original work. Code must be uploaded to GitHub in a folder named JS\_HOT3 for grading.**

***See screen mockup on next page.***

Use the **const or let** keyword to define variables wherever possible.

80% - Well done!

1. Create the web form as shown in the mockup
   1. Use the <header>, <h1>, and <h2> tags for the header \_\_\_\_\_ / 1pts
   2. Use the <form> tag to wrap the form. \_\_\_\_\_ / 2pts
      1. Set the method to POST
      2. Set the action to "cryptoSale.php" (this php script does not exist, but you can demonstrate that the client side coding as if it did.)
   3. Use a <fieldset> tag for each section of the form \_\_\_\_\_ / 2pts
   4. Use a <legend> tag for the header of each section \_\_\_\_\_ / 2pts
   5. Use <label> tags around all control labels \_\_\_\_\_ / 2pts
   6. Use an asterisk (\*) to mark fields that are required \_\_\_\_\_ / 1pt
   7. Use <input> tags to create the shown fields \_\_\_\_\_ / 2pts
      1. 2 radio buttons
      2. 3 check boxes
      3. 12 text fields
   8. Use a <button> tag for the submit button (remember to set the type) \_\_\_\_\_ / 2pts
   9. Use <div> tags to display error messages as shown \_\_\_\_\_ / 2pts
2. Use **Bootstrap** and **CSS** to style the web page
   1. Integrate jQuery via CDN (use the minified version, add the script tags to the bottom of the <body> element) \_\_\_\_\_ / 2pts
   2. Integrate Bootstrap via CDN (add the script tags to the bottom of the <body> element) \_\_\_\_\_ / 2pts
   3. Header with a red background, black text color h1, and white text color h2 \_\_\_\_\_ / 2pts
   4. Header with centered text \_\_\_\_\_ / 2pts
   5. Apply the following Bootstrap classes \_\_\_\_\_ / 6pts
      1. Use the **form** class on the <form>
      2. Use the **form-group** class
      3. Use the **form-control** class for text fields and dropdowns
      4. Use the **form-check-input** classes for radio buttons
      5. Use the **form-check-input** classes for checkboxes
      6. Use the **text-danger** class for error messages
3. Use **JavaScript** to do the following:
   1. Bind a handler to the **submit** event of the form that performs the following tasks:
      1. Validate that the user has selected a type of cryptocurrency.

\_\_\_\_\_ / 10pts

* + 1. Validate that the user has selected at least one type of wallet (user's should be allowed to select both).

\_\_\_\_\_ / 10pts

* + 1. Validate that the user has entered values for all of their Customer Information.

\_\_\_\_\_ / 10pts

* + 1. Validate that the user has entered values for all of their Billing Information.

\_\_\_\_\_ / 10pts

* + 1. Validate that the user has entered the same value in the "Email" and "Confirm Email" boxes.

\_\_\_\_\_ / 10pts

* + 1. When any error on the page occurs the general error message at the top of the page is displayed.

\_\_\_\_\_ / 5pts

* + 1. When an error occurs in one of the 4 unique sections, an error is displayed for that section. \_\_\_\_\_ / 5pts
    2. Display error messages only when there is a problem. Otherwise hide them. \_\_\_\_\_ / 10pts
  1. When the "Same as above" checkbox is checked, populate the Billing Information with the same data that the user entered in the Customer Information section and disable the Billing Information fields.

\_\_\_\_\_ / 5pts

* 1. When the "Same as above" checkbox is unchecked, clear the Billing information and enable the fields.

\_\_\_\_\_ / **5pts**

