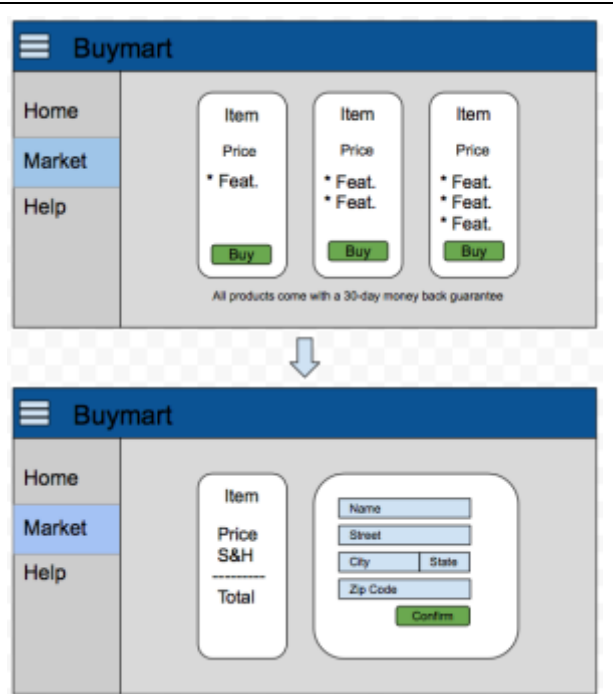


# Exosite UI Coding Challenge

Thank you for applying to join the Exosite Software Products Group. By building a single page app for simple IoT device purchasing, you will have the opportunity to share how you go about solving problems with code. We look forward to seeing your work!

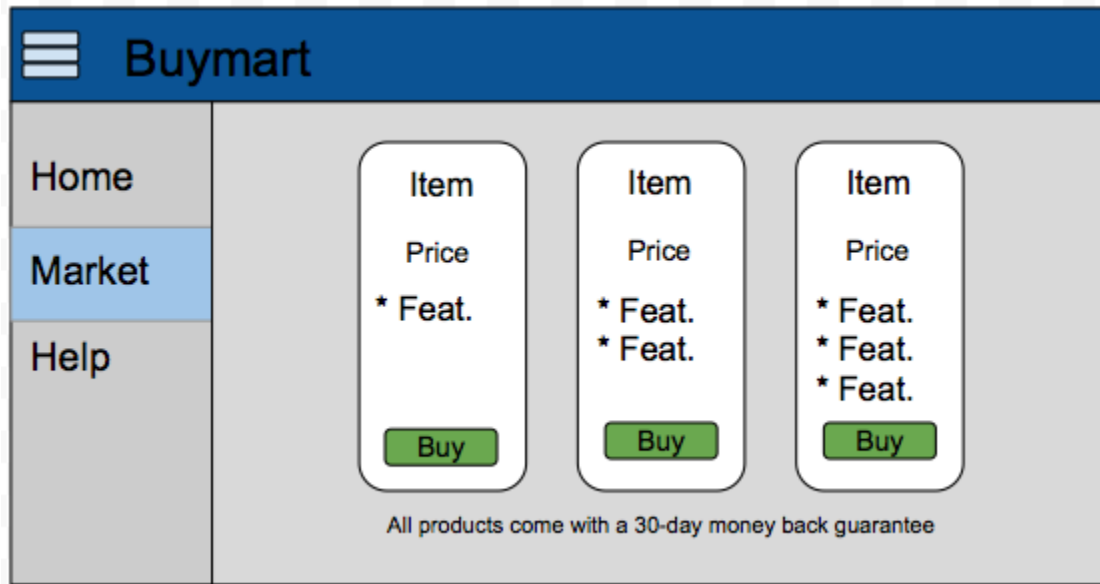
## Simple Product Checkout

1. Your app should show the three products for sale, transitioning nicely to the shipping input form.
2. Use any reasonable combination of Javascript front end frameworks / libraries as you. We use React with ES5 with a visual design based on Material UI, but do not feel pressured to do so yourself. Optimize for personal comfort if anything.
3. Point us to a github repo or send us a tarball/zip file. Please include a README that shows
  - a. How we can run it (e.g. npm install & npm start)
  - b. Describes any design choices and tools
  - c. Honestly assesses room for improvement



## Catalog

The catalog is the first page in the app and is fairly straight forward in presentation.



The data used to load the catalog page is JSON in the form of a list of 1-3 objects. Each object has the following properties.

- *name*: Name of product
- *sku*: The catalog ID of the product
- *price*: Price of product
- *features*: An array of strings describing the product

All content is center aligned and containers should be vertically aligned with each other.

You should fetch this information asynchronously and not statically load it into the page.

Clicking the Buy button should select that item and transition to the next page.

## Shipping Form

The shipping form is made up of two parts: The actual form and the calculator on the left.

The wireframe shows a web interface for Buymart. At the top is a dark blue header with a hamburger menu icon and the text 'Buymart'. Below the header is a vertical sidebar on the left with three buttons: 'Home', 'Market' (highlighted in blue), and 'Help'. The main content area has a light grey background. On the left side of this area is a white rounded rectangle containing a calculator with labels 'Item', 'Price', 'S&H', a dashed line, and 'Total'. To the right of the calculator is another white rounded rectangle containing a form with input fields for 'Name', 'Street', 'City', 'State', and 'Zip Code', followed by a green 'Confirm' button.

The form should be nominally validated to be:

- Completely filled out
- Only valid state abbreviations

The form should clearly indicate which fields have errors and prevent submission if any are present.

For the price calculator, we will include the following S&H fee in addition to the price based on the following structure:

- MN pays no extra fees
- NY, CA, MA pays an extra \$7.50
- GA, AL, FL pay only \$3.99
- All other states pay \$5.99

When the final confirm button is pressed, they should be taken to a Thank you page. Feel free to display whatever you would like.

## Other Parts

There's no need to make Home, Help, or other elements on the page interactive / clickable, but they do need to exist. Otherwise the lines / coloration on for both wireframes are intentional, but any reasonable color palette could be used for the various colors provided:

- The top bar is a dark blue
- The Buy buttons are a shade of green
- The info boxes are white
- The general background is a light grey

## Extra Credit

Feel free to ad lib anything you think would be useful or visually appealing. Some example things to add are included below.

1. **Responsiveness:** Relative spacing should be preserved to 800px.
2. **Mobile View:** For small screens, create an arrangement that could work on a small screen.
3. **Collapsible Side Nav:** Clicking the Hamburger can hide the side nav with no issues.