Overview Questions

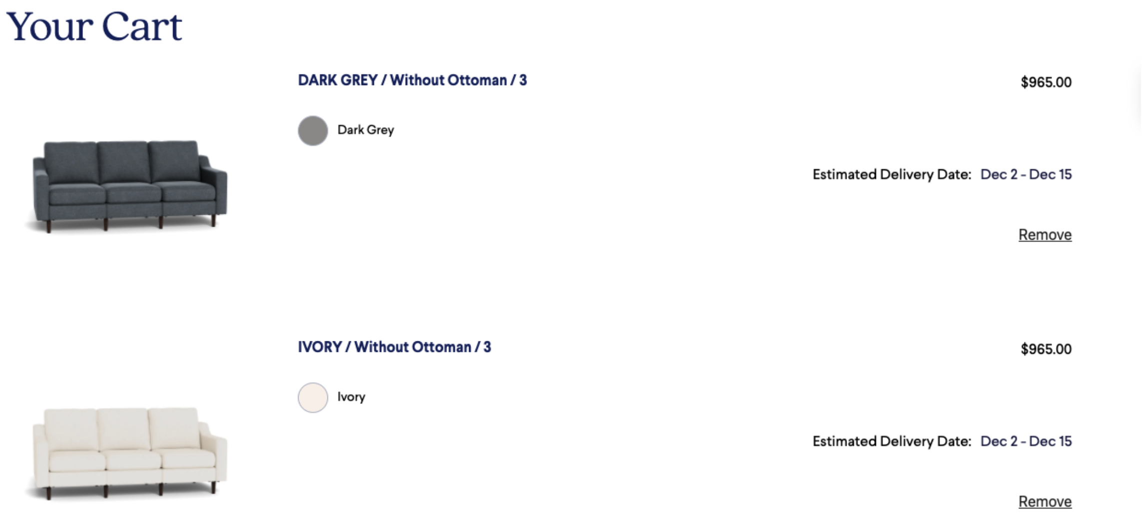
1) What are some programming paradigms
2) What's JSX
3) What's a props
4) What's a state
5) What are refs
6) What's redux / use context
7) What's node
8) What is express js
9) What's a route
10) What's a middleware
11) What's unit testing
12) Name some unit testing frameworks

Technical Interview #1:

Before you began creating a GIthub Repo, and an empty Next JS or React JS / Node project.

Push your first commit only when you are ready to start. You will have 2 hrs MAX to complete. Your first and last commit will be used to measure the length you took to finish the assignment.

1) Save the UI Screenshot below



Your Cart

DARK GREY / Without Ottoman / 3

$965.00

● Dark Grey

Estimated Delivery Date: Dec 2 - Dec 15

Remove

IVORY / Without Ottoman / 3

$965.00

○ Ivory

Estimated Delivery Date: Dec 2 - Dec 15

Remove

2) Copy the following code to the top of App.js / index.js. You will use these variables to create the UI

```
//Styling variables
const BLUE = "#172162"; //"rgb(23, 33, 98)";
const LIGHT_GREY = "#6e7484";
const BLACK = "#000000";


//First part given
const lineItems = [
  {
    id: 1,
    title: "Grey Sofa",
    price: 499.99,
    quantity: 1,
    image:

"https://www.cozey.ca/_next/image?url=https%3A%2F%2Fcdn.shopify.com%2Fs
%2Ffiles%2F1%2F0277%2F3057%2F5462%2Fproducts%2F2_Single_shot_DARK_GREY_
OFF_OFF_SLOPE_17f0f115-11f8-4a78-b412-e9a2fea4748d.png%3Fv%3D1629310667
&w=1920&q=75",
    swatchColor: "#959392",
    swatchTitle: "Grey"
  },
  {
    id: 2,


    title: "Blue Sofa",
    price: 994.99,
    quantity: 1,
    image:

"https://www.cozey.ca/_next/image?url=https%3A%2F%2Fcdn.shopify.com%2Fs
%2Ffiles%2F1%2F0277%2F3057%2F5462%2Fproducts%2F3_Seater_SofaSofa_Ottoma
n_Off_Arm_Configuration_Two_Arms_Arm_Design_Slope_Chaise_Off_Fabric_Nav
y_Blue2.png%3Fv%3D1629231450&w=1920&q=75",
    swatchColor: "#191944",
    swatchTitle: "Blue"


  },
  {
    id: 3,
    title: "White Sofa",
    price: 599.99,
    quantity: 1,
    image:
```

```
"https://www.cozey.ca/_next/image?url=https%3A%2F%2Fcdn.shopify.com%2Fs
%2Ffiles%2F1%2F0277%2F3057%2F5462%2Fproducts%2F2_Single_shot_IVORY_OFF_
OFF_SLOPE_5379af1f-9318-4e37-b514-962d33d1ce64.png%3Fv%3D1629231450&w=1
920&q=75",
    swatchColor: "#F8F1EC",
    swatchTitle: "White"
  },
];


const SUBTOTAL = 2094.97;
const HST = 272.3461;
const TOTAL = 2382.3161;
const ESTIMATED_DELIVERY = "Nov 24, 2021";
```

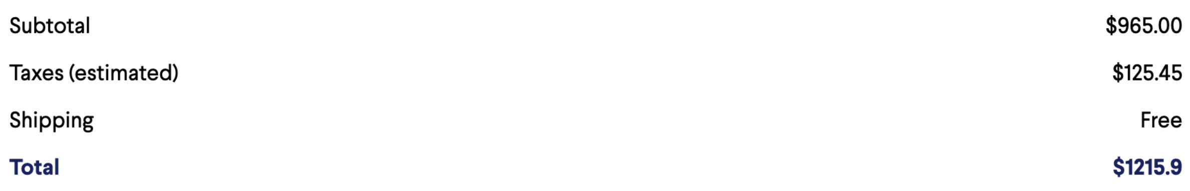
3) Build the UI from step 1.

You may use any resources to complete the task, such as google, github, etc.

NOTE: the "Estimated Delivery Date" on each line item is static for now, and was provided to you in the previous step. (We will make the dynamic later)

4) Create the Cart Fees UI

Underneath the line items, render out the carts total, subtotal, and taxes. These values are statically provided for you already (in Step 2). You simply need to show it in the UI.



Subtotal	$965.00
Taxes (estimated)	$125.45
Shipping	Free
Total	**$1215.9**

5) Move the line items for the constant variable into the state. Set the constant variable provided in the previous section as the default.

6) Create a function called removeLineItem(lineItemId)

This function takes a line item id. It should remove the line item with this id from the line items array in the state. Once you write this function, please connect it to the remove button in the line item (in the UI). When this button is clicked, that specific item should disappear from the UI.

7) Create a function called addLineItem(lineItem)

This function takes a line item (with the object values, as the original line items) and adds it to the line items array. Add a UI Button on the screen, that when you click will add an item to the cart.

8) Create a function called calculateFees

This function takes the line items currently in the cart and dynamically calculates the fees (total, tax, subtotal). Here is the formulas:

- the subtotal is the sum of all the prices of each line item
- the tax is the total, multiplied by the tax rate of 0.13
- the shipping is a $15 flat fee
- the total is the subtotal + tax + shipping

The criteria:

- the function should load the fees correctly on page load (or refresh)
- the function should recalculate the fees when the user removes a line item or adds a line item

9) Create API Route to Get Line Items

Next, we will make the default line items loaded into our app be dynamic. To do this, create an express route that returns the default variants given to you in the prior steps. Before you do this, remove the line items array (given in step 2) from the app. Then create your API route that returns the default line items. Make sure when your app loads or refreshes, these default values are the ones that are rendered onto the screen.

10) Create a Postal Code Input

Create an input in the UI under the Cart Fees (total, subtotal, tax) that takes a user's current postal code. We will use this input in the next step to dynamically set the estimated delivery date for the line items.

11) Refactor the get line items api route to respond with dynamic estimated delivery date per line item

Next, we are going to make the get line items API route send back the estimated delivery date value. To do this you will need some data. Copy this to your api route:

```
const DELIVERY_DATES = [
    {
        postal: "V",
        ids: [2],
        estimatedDeliveryDate: "Nov 24, 2021"
    },
    {
        postal: "V",
        ids: [1,3],
        estimatedDeliveryDate: "Nov 19, 2021"
    },
    {
        postal: "M",
        ids: [2,3],
        estimatedDeliveryDate: "Nov 22, 2021"
    },
    {
        postal: "M",
        ids: [1],
        estimatedDeliveryDate: "Dec 19, 2021"
    },
    {
        postal: "K",
        ids: [1,2,3],
        estimatedDeliveryDate: "Dec 24, 2021"
    },
]
```

This array of objects contains the estimatedDeliveryDate based on the line item id and the FIRST letter of a postal code.

Using this, modify your get line items API route to handle getting the postal code from the user and return an updated line items array that includes the "estimatedDeliveryDate" value per line item, which you can then render onto the UI instead of the static variable we had set previously.

Technical Interview #2 - (frontend focus)

Build this UI to pixel perfection.

https://www.figma.com/file/oEF86Gr6PBaZAMBsIdMbjN/Technical-Frontend-Interview?node-id=0%3A1&t=QKvw6QpPX10Z7Y11-1

Final Note: Please send a loom video of having the app open and going through the code.