NOCTI Prep

## Frontend Requirements

1. One page must be a new order page containing:
   1. A link to the view all orders page
   2. An HTML form with:
      1. The name of the customer who is ordering
      2. The address of the customer who is ordering (shipping location)
      3. The number of each of three items the customer wants to order
      4. A submit button to post the form data to the Create Order POST endpoint
2. One page must be a view all orders page containing:
   1. A link back to the new order page
   2. An HTML table containing the following data for each order that has been submitted:
      1. The order’s customer’s name and address
      2. Each item ordered’s Name, Unit Price, Retail Price, Profit Per Unit, Quantity, Subtotal, and Sales Tax
      3. The order’s shipping costs, and total
   3. The total number of units sold and total profit across all orders

## Handle Create Order POST Endpoint on the Backend

1. Create three item objects with the following information:
   1. Fortnite Battlepass, Unit Price: $7.13
   2. $19 Fortnite Card, Unit Price: $14.28
   3. Early-Access Preorder DLC Elite Edition, Unit Price: $49.78
2. Create an order object with a list of order items, and a subtotal, sales tax, profit, and total all starting at 0.
   1. Add the customer’s name and address as properties of the order object
3. For each item the user could buy
   1. Create a copy or instance of that item, and add that copy to the order items list
   2. Add a quantity property to that item whose value is the number of that item the customer ordered
4. For each item in the order items list:
   1. Calculate the Retail Price and Profit and add them as properties to the item
      1. Profit is 40% of the unit price
      2. Retail Price is Profit plus Unit Price
   2. Calculate the subtotal of the item and add it as properties to the item
      1. Multiply the Retail Price by the Quantity of the item
   3. Calculate the sales tax of the item and add it as properties to the item
      1. Sales tax is 6% of the subtotal
   4. Update the totals of the order
      1. Add the subtotal of the item to the order’s subtotal
      2. Add the sales tax of the item to the order’s sales tax
      3. Add the profit of the item to the order’s profit
5. Calculate the shipping costs
   1. If the subtotal of the order is less than 40, the order’s shipping property is 15
   2. Otherwise, if the subtotal of the order is less than 150, the order’s shipping property is 10
   3. Otherwise, the shipping is free
6. Calculate the order’s Total
   1. Add together the order’s subtotal, sales tax, and shipping cost
   2. Round each property of the order with a dollar amount to the nearest penny
7. Add this order to the master list of orders to be read by the view all orders page
8. Redirect the user to the view all orders page to verify the new order

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Backend Code  |  | | --- | | Setup Server and Program Variables Import Modules  Create Constants and Global Variables  Start HTTP Listen Server with ExpressJS | | Handle GET request to root Use EJS to render the new order page | | Handle POST request from new order page Handle the Create Order as described  Redirect user to show all orders GET endpoint | | Handle GET request to view all orders Send all orders data to view all orders template | | Front End TemplatesNew Order Page  View All Orders Page You will need to use EJS to build the tables and populate the data |

## Required Skills

In order to pass the NOCTI, you need to be able to do the following from memory:

1. Work with Javascript Objects, basic calculations, arrays, and loops.
2. Create a NodeJS application
3. Use ExpressJS with an HTTP Listen Server
   1. Must be able to handle GET and POST endpoints
   2. Must be able to handle Query Parameters and HTML Form data
4. Create and serve EJS templates
   1. Must be able to send data to the template engine
   2. Must be able to use embedded Javascript in the EJS template to create HTML
5. Create an HTML table using Javascript and populate it with data
   1. This will be done using embedded Javascript in an EJS template
6. Create a cohesive website with a functioning service
7. Create a flowchart or pseudocode list of a program’s functions