

**Exam Premise:**

Create an EJS application that utilizes the following given REST API, which returns **shopping carts**, to calculate **average cart prices**.

**Exam Details:**

The following endpoint can be used to request list of **20 shopping carts**:

<https://dummyjson.com/carts>

You will receive a response similar to this, which for humans is pretty illegible:

```
{
  "carts": [
    {
      "id": 1,
      "products": [
        {
          "id": 59,
          "title": "Spring and summershoes",
          "price": 20,
          "quantity": 3,
          "total": 60,
          "discountPercentage": 8.71,
          "discountedPrice": 55
        },
        {
          "id": 88,
          "title": "TC Reusable Silicone Magic Washing Gloves",
          "price": 29,
          "quantity": 2,
          "total": 58,
          "discountPercentage": 3.19,
          "discountedPrice": 56
        },
        {
          "id": 18,
          "title": "Oil Free Moisturizer 100ml",
          "price": 40,
          "quantity": 2,
          "total": 80,
          "discountPercentage": 13.1,
          "discountedPrice": 70
        },
        {
          "id": 95,
          "title": "Wholesale cargo lashing Belt",
          "price": 930,
          "quantity": 1,
          "total": 930,
          "discountPercentage": 17.67,
          "discountedPrice": 766
        },
        {
          "id": 39,
          "title": "Women Sweaters ....."
        }
      ]
    }
  ]
}
```

Use a **JSON formatter** to better visualize your response (for instance [here](#) or [here](#)):

```
1  {
2    "carts": [
3      {
4        "id": 1,
5        "products": [
6          {
7            "id": 59,
8            "title": "Spring and summershoes",
9            "price": 20,
10           "quantity": 3,
11           "total": 60,
12           "discountPercentage": 8.71,
13           "discountedPrice": 55
14         },
15         {
16           "id": 88,
17           "title": "TC Reusable Silicone Magic Washing Gloves",
18           "price": 29,
19           "quantity": 2,
20           "total": 58,
21           "discountPercentage": 3.19,
22           "discountedPrice": 56
23         },
24         {
25           "id": 18,
26           "title": "Oil Free Moisturizer 100ml",
27           "price": 40,
28           "quantity": 2,
29           "total": 80,
30           "discountPercentage": 13.1,
31           "discountedPrice": 70
32         },
33         {
34           "id": 95,
35           "title": "Wholesale cargo lashing Belt",
36           "price": 930,
37           "quantity": 1,
38           "total": 930,
39           "discountPercentage": 17.67,
40           "discountedPrice": 766
41         },
42         {
43           "id": 39,
44           "title": "Women Sweaters ....."
45         }
46       ]
47     },
48     {
49       "id": 2,
50       "products": [
51         {
52           "id": 96,
53           "title": "lighting ceiling kitchen",
54           "price": 30,
55           "quantity": 2,
56           "total": 60,
57           "discountPercentage": 14.89,
58           "discountedPrice": 51
59         },
60         {
61           "id": 91,
62           "title": "Black Motorbike",
63           "price": 569,
64           "quantity": 3,
65           "total": 1707,
66           "discountPercentage": 13.63,
67           "discountedPrice": 1474
68         },
69         {
70           "id": 9,
71           "title": "Infinix INBOOK",
72           "price": 1099,
73           "quantity": 1,
74           "total": 1099,
75           "discountPercentage": 11.83,
76           "discountedPrice": 969
77         },
78         {
79           "id": 16,
80           "title": "Hyaluronic Acid Serum",
81           "price": 19,
82           "quantity": 1,
83           "total": 19,
84           "discountPercentage": 13.31,
85           "discountedPrice": 16
86         }
87       ]
88     }
89   ]
90 }
```

array of 20 shopping carts

first cart (products are collapsed)

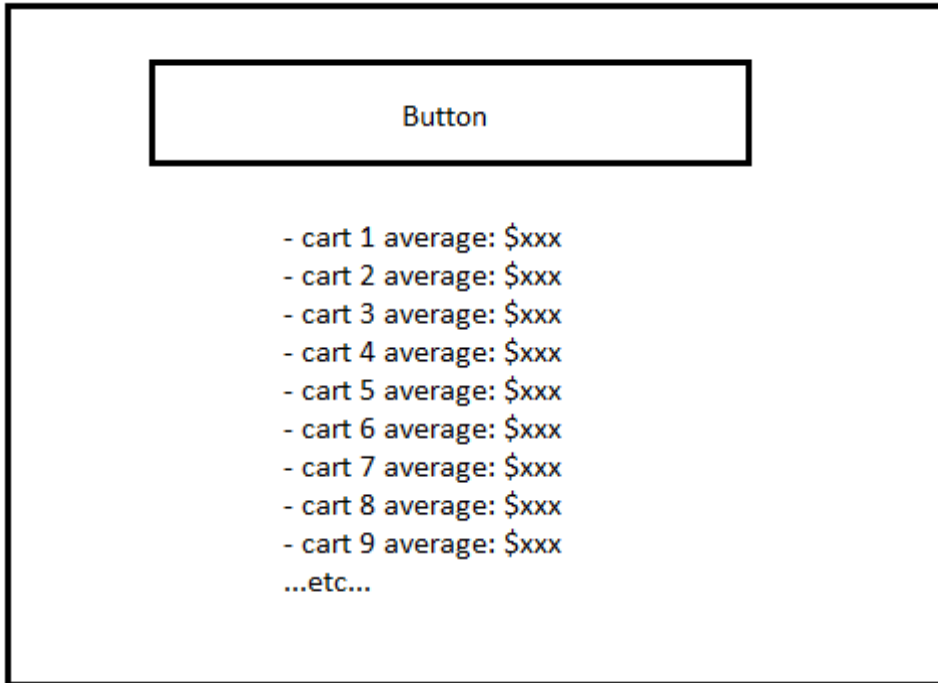
second cart (products are expanded)

## UI REQUIREMENTS:

When the page loads, there should be nothing on the page except **one button** that says: “**Calculate cart averages**”. When the user clicks that button, the page should **make the API** call to **retrieve the 20 shopping carts**. You then need to calculate the **average price of every cart** and display it. You can choose to display it on the same page underneath the button, or on a new page. You can use any html controls of your choice.

**NOTE:** Every cart has several items in its '**products**' array. Every **product** has a **quantity**. We need the average of every cart considering how many products we have, and what **quantity of every product**.

Here is a sample UI mockup for reference:



## Technical requirements:

Use **EJS** as your JavaScript Framework, and utilize the libraries learned in the course such as **express**, and **axios**. You're welcome to use additional libraries as you choose.

You can either create a completely **new EJS project**, or you can use one of the **EJS templates** we have shown in class. If you do use a template, **remove any code** from it that's not being used from the project. **Failing to do so will result in points deduction.**

## Extra Credit (10 points):

Use **CSS** to **Expand/Collapse** the cart averages, as shown in class when we were discussing bootstrap. The bootstrap sample page has this functionality demonstrated in the section called “**Collapse**”.

**Submission:** Zip your folder **excluding the node\_modules** folder and submit it on **Blackboard**. You can submit **as many times as you want**, only the **last submission will be graded**. **Submit often**. The exam automatically **closes at 11.59pm**. **Only code submitted by the deadline will be graded**.

**No exceptions for late submissions will be accepted.**

Make sure your project **compiles and runs**. If the project doesn't run, you will forfeit points.

**Do not forget to comment and reference where needed and appropriate.**

**Zero tolerance in plagiarism and collaboration. Violation thereof will result in a zero of the assignment for all involved parties, as well as a case report to the department.**

## Grading Rubric:

Item	Points
UI setup in EJS	30
API call	10
JSON parsing & calculation	60
<b>Total (w/ Extra Credit)</b>	<b>100 (110)</b>