**Learn JavaScript by Building a Food Ordering System**

**🔹 Part 1: Introduction**

* What you will build: A shopping cart for food.
* What you will write: JavaScript to make the page interactive.

**🔹 Part 2: Step-by-Step JavaScript Guide**

**✅ Step 1: Create an Empty Cart**

let cart = [];

**🔹 Explanation:**

This is like your **shopping basket** — it starts empty.Whenever the user clicks “Add”, we put item details into this basket (array).

Each item in the cart has:

1. **name** – the name of the food
2. **price** – how much it costs
3. **quantity** – how many of that item the user added
4. ✅ (Optional) **index** – we don't store it directly, but we use its position in the array to find or update the item

So it object.

**🔹 Part 2: JavaScript Steps (Quick Guide)**

**Handle “Add” Button Clicks**

document.querySelectorAll('.add-to-cart').forEach(button => {

button.addEventListener('click', () => {

// 1. Find the item

// 2. Get name and price from HTML

// 3. Add to cart or increase quantity

// 4. Call updateCart()

});

});

**🔍 Explanation of Methods**

* querySelectorAll() – selects all elements with the class .add-to-cart(see html file)
* forEach() – runs code for each button(for each food)
* addEventListener() – listens for clicks on buttons(like “click” , “mousemove”…..)

**Get the Item Info**

const itemElement = button.closest('.item');

const itemName = itemElement.getAttribute('data-name');

const itemPrice = parseFloat(itemElement.getAttribute('data-price'));

### 1 closest ('.item')

Finds the **nearest parent** element (or the element itself) that has the class .item.  
Used to locate the full food item box when a button inside it is clicked.

### 2️ getAttribute('data-name')

Gets the value of an attribute from the HTML element.  
Here, it reads the data-name or data-price like this:

<div class="item" data-name="Firfir" data-price="60">

### 3️ parseFloat()

Converts a string (like "60") into a number (60.0).

✅ **Step 4: Add Item to the Cart Array**

const existingItem = cart.find(item => item.name === itemName);

if (existingItem) {

existingItem.quantity += 1;

} else {

cart.push({ name: itemName, price: itemPrice, quantity: 1 });

}

Explain: If the item is already in the cart, just increase the quantity. Otherwise, add it.

**✅ Step 5: Create the updateCart() Function**

🧠 This function keeps the cart area updated whenever something changes.

**🛠️ What it does:**

* Clears old cart items
* Loops through each item
* Displays item info + total
* Adds ➕ ➖ 🗑️ buttons
* Updates total price

**🧩 Sample Code (Guess and Complete It!)**

function updateCart() {

const cartItemsElement = document.querySelector('.cart-items');

cartItemsElement.innerHTML = ''; // clear old cart

let totalPrice = 0;

cart.forEach((item, index) => {

// create cart-item div

// show item.name, item.quantity, item.price

// create +, -, 🗑️ buttons

// update total price

totalPrice += item.price \* item.quantity;

});

document.getElementById('total-price').textContent = totalPrice.toFixed(2);

}

📝 Try filling in:

* How to create HTML for the cart item
* How to add event listeners for the buttons
* How to increase, decrease, or delete items

**✅ Step 6: Add “Cancel All” and “Place Order” Buttons**

🟥 **Cancel All Button**  
When this button is clicked:

* Clear the cart (set it to an empty array)
* Call updateCart() to refresh the display

🟩 **Place Order Button**  
When this button is clicked:

* If the cart is empty → show an alert: "Your cart is empty!"
* If not → show the total price in an alert like: “Order placed! Total: $\_\_\_”
* After placing the order, clear the cart and update the screen

Let me know if you want to include this in the PDF or continue with another step!

**🧠 Bonus Tips:**

* Use console.log() to debug.
* Test each part before moving to the next.
* Think about what happens when someone adds the same item again.

**🎁 Challenge:**

**✅ Your Task**

Use this guide to write your JavaScript step by step.  
Ask yourself:

* What happens when I click “Add”?
* What does the cart array look like?
* Try to add your own item to the menu.
* Add a feature to confirm before placing an order.