**4. Development Phase**

**Objective:**

To convert the planned design and logic into a functional website using code. This phase involves building each web page, styling it, and implementing interactivity using JavaScript.

**Overview of Development Approach**

Since you're working on a **front-end-only project**, development was focused on:

* Writing HTML to structure content
* Applying CSS for visual styling and responsiveness
* Using JavaScript for dynamic features like "Add to Cart" and total price calculation

The development process followed a **step-by-step page-building approach**, ensuring that each component was tested before moving to the next.

**Technology Stack Used**

| **Purpose** | **Technology** |
| --- | --- |
| Structure (markup) | HTML5 |
| Styling | CSS3 |
| Interactivity | Vanilla JavaScript |
| Hosting | GitHub Pages / Netlify |
| Code Editor | VS Code |
| Version Control (optional) | Git + GitHub |

**Page-by-Page Development**

**Home Page (index.html)**

* Header with logo or site name
* Welcome message or banner
* Navigation bar (links to Menu, Contact, Cart)
* "Order Now" button linking to Menu page

**Menu Page (menu.html)**

* Structured layout of food items using HTML
* Each item includes:
  + Image
  + Name and price
  + “Add to Cart” button
* CSS applied to make cards visually appealing and responsive using Flexbox or Grid
* JavaScript function (addToCart()) connected to each button

**Cart Page (cart.html) *(optional or dynamic section on same page)***

* Displays list of selected items
* JavaScript dynamically updates item list and price
* Button to "Place Order" triggers confirmation alert or popup

**Order Confirmation (could be an alert or dedicated section)**

* After placing the order, user sees a message like:

Thank you for your order! Your food will be delivered soon.

**JavaScript Functionality Highlights**

| **Feature** | **JavaScript Implementation** |
| --- | --- |
| Add to Cart | Adds selected item to an array or localStorage |
| Show Cart Items | Dynamically creates HTML elements to show cart contents |
| Total Price Calc | Loops through cart array and adds prices |
| Confirmation Message | Shows alert or updates DOM with confirmation message |

**Example:**

javascript

CopyEdit

let cart = [];

function addToCart(itemName, itemPrice) {

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

alert(itemName + " added to cart!");

}

function calculateTotal() {

let total = 0;

cart.forEach(item => total += item.price);

alert("Total: $" + total);

}

**CSS Styling Highlights**

* Consistent color theme (e.g., warm food tones: red, orange, cream)
* Hover effects on buttons
* Font styling using Google Fonts
* Responsive layout using:
  + Media queries
  + Flexbox or Grid system
* Fixed navbar for easy navigation

**Challenges Faced During Development**

| **Challenge** | **Solution Applied** |
| --- | --- |
| JS not updating cart properly | Used console.log() for debugging |
| Styling breaks on small screens | Applied media queries for responsive design |
| Buttons not triggering functions | Double-checked event listeners and IDs |
| Repetition in HTML for menu items | Created reusable HTML structure per card/item |

**Summary**

The Development Phase transformed design concepts into an actual, working food ordering website using HTML, CSS, and JavaScript. Each webpage was carefully built and styled, ensuring the layout was clean and user-friendly. JavaScript provided the necessary logic to allow basic ordering features like adding to cart and calculating the total amount.

Despite being a small project, this phase involved writing clean, organized code and solving real-world problems like responsive design and user interactivity. It demonstrates how even beginner-level web technologies can be used effectively to create a functional mini-application when combined with thoughtful planning and design.