**3. System Design Phase**

**Objective:**

To create a blueprint for how the website will function and look before starting actual development. This includes page layout, navigation flow, user interaction, and how basic data (like menu items and cart contents) is managed on the front end using JavaScript.

**Website Architecture**

Your website is likely **multi-page**, and structured something like this:

index.html → Home Page

menu.html → Menu Items Page

cart.html → Shopping Cart (optional)

contact.html → Contact Info / Order Confirmation

style.css → All CSS styles

script.js → JS functions (Add to Cart, Show Total, etc.)

images/ → Folder for food images/icons

**UI/UX Design (Wireframes / Layout Plan)**

Before writing code, it's helpful to plan how each page looks. Here's a basic layout breakdown:

**Home Page (index.html)**

* Website title/logo
* Welcome message or restaurant introduction
* "Order Now" button leading to Menu page

**Menu Page (menu.html)**

* Categories: e.g., Pizzas, Burgers, Beverages
* Each food item with:
  + Image
  + Name
  + Price
  + "Add to Cart" button (calls a JS function)

**Cart Page (optional)**

* List of added items
* Quantity and total price calculation (done in JS)
* "Place Order" button → confirmation alert

**Contact/Confirmation Page**

* Contact form or message (optional)
* Final confirmation message after ordering

**Navigation Flow (How Users Move Through the Site)**

[Home Page]

↓ (Order Now)

[Menu Page]

↓ (Add Items)

[Cart Page]

↓ (Click "Order")

[Thank You / Confirmation Page]

Users should be able to:

* Move back and forth between pages using a fixed top navigation bar
* View cart and modify order before confirming

**Frontend Logic Design (JavaScript)**

Although you don’t have a backend or database, **JavaScript handles user interaction and cart functionality**:

**Key JS Features:**

* **addToCart() Function**: Adds selected item to a temporary cart (array or localStorage)
* **Cart Item List**: Displayed on the Cart Page using DOM manipulation
* **calculateTotal()**: Sums prices of all items
* **placeOrder()**: Shows a thank you message via alert() or modal popup

**Example:**

javascript

CopyEdit

let cart = [];

function addToCart(itemName, itemPrice) {

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

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

}

**Data Handling**

Since there's no backend:

* All **menu data** is hardcoded in HTML or JS
* All **cart data** is stored in:
  + JavaScript arrays OR
  + localStorage (if you want to persist it across pages)

**Visual Design Decisions (CSS)**

* Layout: Flexbox/Grid used for responsive cards
* Fonts: Google Fonts for clean UI
* Colors: Soft food-friendly color palette (e.g., orange, red, white)
* Responsiveness: Use media queries to make site mobile-friendly

**Deliverables from System Design Phase:**

| **Item** | **Description** |
| --- | --- |
| Page Wireframes | Rough layout for each page (drawn or digital) |
| Navigation Flow Diagram | Arrows showing user movement between pages |
| Folder/File Structure | How code files and images are organized |
| JavaScript Functions Plan | List of all core functions like addToCart() |
| UI Design Elements | Buttons, cards, nav bar styling |

**Summary**

The System Design phase serves as a critical bridge between planning and actual development. It transforms ideas and requirements into a structured blueprint that guides the website's construction. Even though this project involves a relatively simple food ordering website with no complex backend, system design ensures that every part of the website — from the layout to interactivity — is planned out and intentionally crafted.

This phase clarified how users will interact with the site, how information will be presented, and how JavaScript will handle basic tasks like adding items to the cart or calculating the total. It also addressed the navigation structure of the website, ensuring smooth transitions between pages like the Home, Menu, Cart, and Order Confirmation.

In this project, system design focused on:

* Designing an intuitive and clean user interface, using HTML for structure and CSS for layout and styling.
* Creating a simple, user-friendly navigation system to move between pages logically (e.g., Home → Menu → Cart → Confirm).
* Outlining the necessary JavaScript functions, such as adding items to a cart, showing alerts, and calculating totals—all done without a backend.
* Organizing files and folders logically, which improves maintainability and readability of the codebase.
* Planning responsiveness, ensuring the website is usable on both desktops and mobile devices using CSS media queries and flexible layouts.

By completing the design phase, the project gained a clear technical direction and visual goal, making the development phase smoother and more efficient. It allowed for early detection of potential design issues and gave a visual reference that could guide consistent and clean implementation.

Whether the project is a small static site or a full-scale application, the System Design phase ensures that the end result is functional, user-friendly, and aligned with project goals.