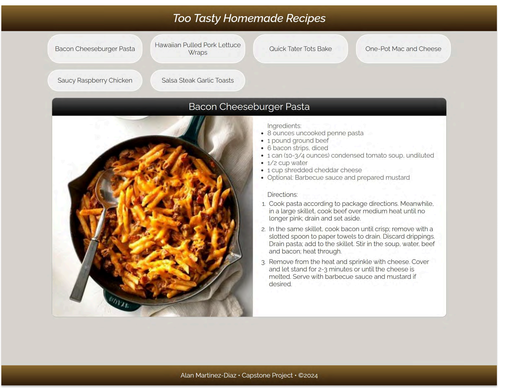
Too Tasty Homemade Recipes: A Bite-Sized Web Experience

Building a simple, fast, and visually inviting recipe website that skips the fluff and serves users exactly what they came for.

BY ALAN MARTINEZ-DIAZ



Introduction

For this project I built a website called Too Tasty Homemade Recipes where I designed and coded everything myself using HTML CSS and JavaScript through Visual Studio Code. The goal of this assignment was to create a functioning website that felt simple inviting and easy to use. The problem I wanted to solve was how to create a recipe website that gives users what they actually come for, the recipes, without forcing them to scroll through long stories or large articles before seeing the ingredients and directions. I wanted to design something minimal yet organized where every recipe would be consistent and clear. My focus was on structure clean code and making sure the site was easy to expand in the future if more recipes were added.

The Process

Developing in Visual Studio Code

Since this was a hard coded project I didn't rely on sketches or prototypes but instead worked directly in code. I started with a basic HTML layout that included a header navigation section a main content area and a footer. I then styled everything using CSS to create a soft warm look that felt approachable. I used Google Fonts pairing Manrope for paragraph text and Teko for bold titles to give the site a friendly and modern personality. I added gradients to the header and footer to create depth while keeping the background light and easy on the eyes.

The functionality came from JavaScript where I stored each recipe inside an array as an object containing the name image ingredients and directions. From there I used JavaScript to dynamically create navigation buttons and render each recipe card when clicked. This meant I didn't have to write multiple HTML pages for every recipe. Everything was built to load within one page making the website fast and easy to navigate.

HTML Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Too Tasty Homemade Recipes</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Manrope:wght@200..800
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
    <link rel="stylesheet" href="css/normalize.css">
    <link rel="stylesheet" href="css/small.css">
    <link rel="stylesheet" href="css/medium.css">
    <link rel="stylesheet" href="css/large.css">

</head>
<body>
    <header><h1>Too Tasty Homemade Recipes</h1></header>
    <nav></nav>
<main>
    <div id="card"></div>
</main>
<footer>Alan Martinez-Diaz &bull; Capstone Project &bull; ©2024</footer>
    <script src="js/scripts-js" type = "module"></script>
</body>
</html>
```

This code sets up the foundation of the entire website. The header displays the site title while the navigation section is where the recipe buttons are dynamically inserted using JavaScript. The main element serves as the live content area where each recipe card is rendered when a user selects one from the menu. Finally the footer updates automatically with the current year using a small JavaScript function. This layout keeps the site organized and clean while ensuring all key content stays on one page.

Recipe Data

```
export const recipes = [
  {
    "recipe": "Bacon Cheeseburger Pasta",
    "photo": "bacon.jpg",
    "ingredients": [
      "8 ounces uncooked penne pasta",
      "1 pound ground beef",
      "6 bacon strips, diced",
      "1 can (10-3/4 ounces) condensed tomato soup, undiluted",
      "1/2 cup water",
      "1 cup shredded cheddar cheese",
      "Optional: Barbecue sauce and prepared mustard"
    ],
    "directions": [
      "Cook pasta according to package directions. Meanwhile, in
      "In the same skillet, cook bacon until crisp; remove with a
      "Remove from the heat and sprinkle with cheese. Cover and l
    ]
  },
```

Each recipe is stored as an object inside an array, making it easy to add, remove, or update recipes without ever touching the HTML. Each object includes the recipe name, image path, alt text, ingredient list, and step-by-step directions. This method makes the website data-driven, meaning the structure of the site doesn't change but only the content does. This keeps the code clean and reusable while making it easy to maintain.

JavaScript Rendering Function

```
//function show food
function showFood(food) {
console.log(food)

let foodSection = document.createElement("section")
let foodRecipe = document.createElement("h2")
let foodPhoto = document.createElement("img")
let foodIngredients = document.createElement("ul")
let foodDirections = document.createElement("ol")

foodRecipe.textContent = food.recipe
foodPhoto.src = `images/${food.photo}`
foodPhoto.alt = food.recipe
foodDirections.innerHTML = `Directions:`
foodIngredients.innerHTML = `Ingredients:`
food.ingredients.forEach(ingredient => {
    let theIngredient = document.createElement('li')
    theIngredient.textContent = ingredient
    foodIngredients.appendChild(theIngredient)
})
food.directions.forEach(direction => {
    let theDirection = document.createElement('li')
    theDirection.textContent = direction
    foodDirections.appendChild(theDirection)
})
```

This function is the core of the site's interactivity. It finds the recipe that matches the user's selection and then dynamically builds an HTML structure to display it. The content inside the <main> element changes every time a new recipe button is clicked. By using JavaScript template literals, the ingredients and directions are looped through automatically to create list items without having to manually write them. This made it easy to expand the project because the same logic works for any number of recipes added to the data array.

Navigation Generation

```
//nav items
recipes.forEach(food => {
    console.log(food)
    const myBtn = document.createElement('button')
    myBtn.textContent = food.recipe
    myBtn.addEventListener('click', () => showFood(food))


    myNav.appendChild(myBtn)
})
```

This small but powerful section of JavaScript creates navigation buttons for each recipe automatically. When the page loads, the code loops through all the recipes in the data array and makes a button for each one. Clicking a button triggers the showFood(food) function, replacing the content in the main section with that recipe's details. This feature made the site feel more dynamic and interactive, even though it was fully built with static code and no frameworks.

Together, these pieces of code form a small but complete ecosystem that powers the entire website — from structure and styling to content and interaction.

Styling

The site uses a bright and soft color palette to keep the focus on the food while maintaining readability. Rounded corners generous spacing and subtle gradients were used to give the design warmth and depth. I made sure that everything looked balanced across mobile and desktop by using responsive CSS breakpoints. This helped the layout adapt naturally without breaking the structure.

Conclusion

Looking back on this project I am really proud of what I built completely from code. This assignment gave me a lot of creative freedom and helped me strengthen my understanding of HTML CSS and JavaScript. I was especially happy with how clean and organized the code turned out and how smooth the recipe navigation feels. My favorite feature I added was the automatic loading of recipes from the JavaScript array because it made adding new dishes fast and simple. If I could go back and improve something I would add a search bar and a favorites option to make the site even more interactive. Overall I feel that I solved my original problem of designing a recipe website that keeps things short and straightforward while still looking appealing and functional.