CSC372: Web Development
Assignment 3: Dine On Campus
Due: Tuesday, February 18, 2025

In this assignment, you will practice layout and simple event handling by creating a website in the style of a blog. You have creative freedom, but this website should be about food and drink recommendations on or close to our campus. Include a minimum of 3 unique recommendations for food establishments within a half-mile radius from Petty building. Your web page should include semantic layouts with at least the following: header, main, section, article, and aside. If other layout components make sense, feel free to add them. Appearance and content are up to you, subject to the following specs:

1. Your main page should include an introduction section that lays out the purpose of the rest of the article. For example, “Are you a culinary adventurer in search of the perfect meal on campus? Look no further! We've curated a list of the most exceptional dishes that are sure to tantalize your taste buds. From fresh pizzas to mouthwatering sushi, this list has got you covered. So, put on your foodie hat and let's dive into the delightful world of flavors.” This introduction should have a minimum of 5 sentences.
2. For each establishment that you recommend, provide a short description (location, specialties, hours, etc.) and include 3 specific dishes that you recommend from there. You will need to have pictures for each of the dishes (they do not have to come from the actual location itself. If you use images from the internet, be sure to give credit for each of them!). Use a flex container that is responsive to the size of the screen. Design your page such that when a user clicks on a specific image, the image becomes bigger than the other 2 like the gallery example from class. This action should also generate a description of the selected dish and how much it costs. This should NOT be a pop-up, but an actual section on the page. There should only be one of these dish descriptions visible at any one time, depending on what image the user clicked on. When the page first loads and the user has not clicked on any image yet, no dish description should be displayed, and all images should be small.
3. Implement a second page that lets users create a meal plan for their day based on your recommendations. This should display a list of your recommended dishes to the left of the page, and the meal plan to the right. If a user selects a dish, add it to the meal plan and display the total dollar amount for their selected meals. The meal plan page does not need to have images or descriptions, just dish names and costs are sufficient. For each dish added to the meal plan, give the user an option to remove it from the plan, or add more orders of it to the meal plan. The meal plan page will not do anything special except show the user a what-if scenario of getting the selected recommended dishes. Make sure you include a link from your main page to the meal plan page.
4. For article sections, be creative and use interesting information about the locations or any promotions going on, if you tried a specific dish recently (it’d be cool if you included your pictures) or information like using flex dollars at these locations.
5. Select color themes and fonts that go well together and avoid using super bright neon colors. Any images appearing on your page should never have a dimension larger than 300px at any time.

To submit your work, you will use your GitHub pages site again. In your *<username>.github.io* repository location that contained the subfolder for hw2, create a subfolder for this homework and name it hw3. All your work for this homework must be contained within this subfolder (HTML, CSS, JavaScript, images). Organize your files so they are easy to locate in subfolders. Do not directly use styles and scripts in the HTML file. All styles should be in .css files, and all scripts should be in .js files.

Be sure to edit the homepage for your GitHub Pages site to include a link to this subfolder, just like you did with hw2.

When you push your changes to your github.io repo, your new sub-folder will also be deployed to your GitHub pages website.

Submit on Canvas: the link to your deployed home page (your-username.github.io). Clicking on your submission should take you to an actual webpage, not a GitHub repository.