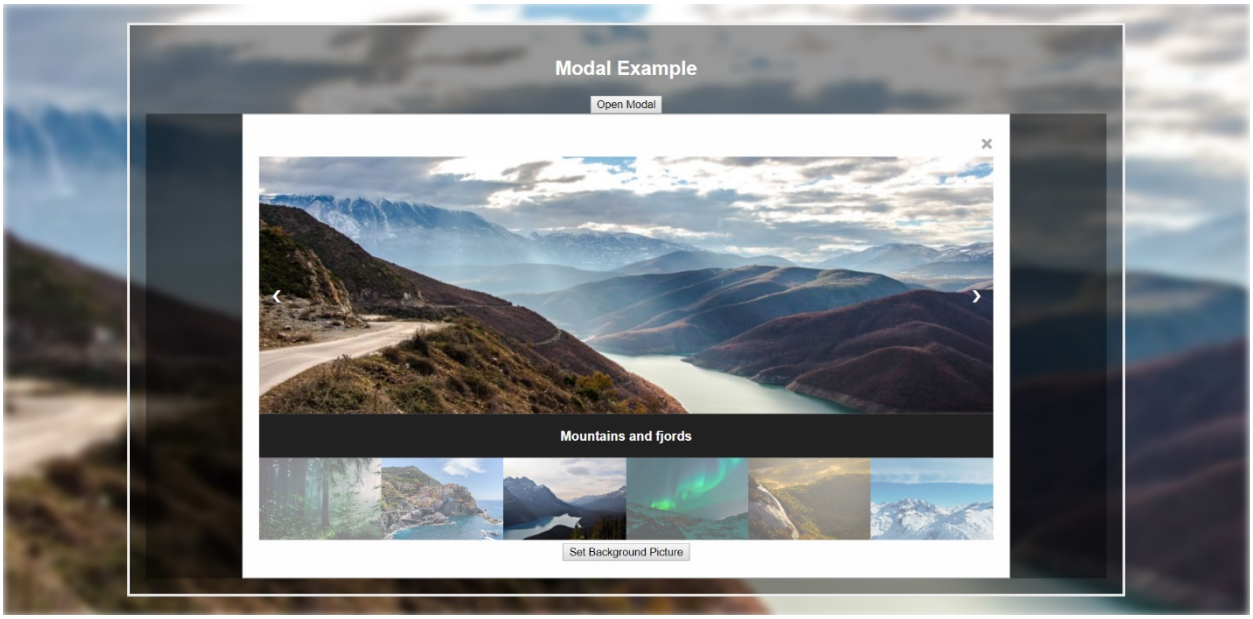
Trung Viet Nguyen

CIS 3308

Tutorial Proposal

I plan to create a framework that allows a pop-up modal to display an image slideshow, from which the user can view the images and select one as the background photo of the page. I came up with this framework by combining different tutorials from w3schools.com (Modal Popup, Image Slideshow, CSS background-image property, etc.). In my current proof of concept code, the JavaScript functions work by injecting properties to CSS using the IDs provided by the DOM elements. The goal is to incorporate this framework into my own website, which is a gaming website and will look nicer if there are multiple background images to choose from. Below is a screenshot of my demo.html page.



My next step is to convert the current code into a Consumer/Provider style framework, where HTML developer can pass in the image paths to the function and have the image slide show constructed for them. The developer can also choose which DOM element is used to trigger the pop-up modal (in the proof of concept code, this is assigned to the Open Modal

button). The framework comes with a CSS file so that the user can change the styling of the framework if necessary.

Currently, most of the code to construct the contents of the pop-up modal is inside the HTML file, and the JavaScript function will change the “display” properties in the CSS file to specify which images are being displayed. I plan to push the code to the JavaScript file to make it easier for the consumer. One idea is to pre-construct the modal in the JavaScript file using the parameters the user put in, then invoke the pop-up once the assigned DOM element is clicked on. Another thing to work on is the blur/opacity value which is currently preset in the CSS file and should also be adjustable by the user.

Finally, I plan to include some validate codes to make sure the user puts in the correct parameters. However, I will spend less time on this part and will probably only include some basic error-catching methods, as the user is also responsible for their own function invocation.