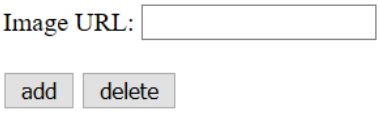
CSCI 452/553: Network and Web Programming

Javascript Programming Project: Image Planner

Due April 16th, 11:59:59 PM

- All development must be done with Github. Specifically, there must be at least one commit for every ten (or fewer) lines of code (excluding empty lines and lines with only comment). With each commit, the log must explain the purpose of the added/changed code. The Github repo must remain private before the deadline and public after the deadline.
- Submit a single plain text file named repo.txt at http://hucs.dynu.net/lij/courses/submit_hw.html
  The text file should include only the URL of your Github repo.
- Grading
  All or significant portion of the points will be deducted at the discretion of the instructor if any of the above requirements are not met. Plagiarism will always lead to zero credit for this assignment and potentially for the whole course. No credits will be given if the repo is modified after the deadline. Other than that, the points will be given based on the correctness.

This assignment tests your understanding of JavaScript and its interaction with HTML user interfaces and DOM. You should create an image.html that have a blank area (named planning area) followed by the following controls:



When user enters a URL in the input box and clicks the 'add' button,

- If the URL is valid in terms of format, the image at the URL should be added to the planning area. If the image size if larger than 100x100, display it as no larger than 100x100. Fill a row with images first before putting images in another row.
- If the URL has an invalid format, an error message should be displayed under the controls.

When user enters a URL in the input box and clicks the 'delete' button,

- If the URL is valid in terms of format,
  - If the image at the URL is in the blank area, remove it.
  - Otherwise, show an error message under the controls.
- If the URL has an invalid format, an error message should be displayed under the controls.

When user clicks an image in the planning area, its URL should show up in the input box.

When user hovers the mouse on an image, the image should be zoomed in to be the lesser of 500x500 and

original size. After the cursor moves away from the image, the zooming effect should disappear.
Any error message should disappear when user enters a new URL or click either button.

In total you will turn in the following files:

- **image.html**, your web page
- **image.css**, the style sheet for your web page
- **image.js**, the JavaScript code for your web page
- A plain text file named readme.txt that contains the following items:
  - The URL of your project on a hosting service.
  - The URL of your project on Github.
  - The date and time, brief description in one or two sentences, and the URL of the most significant commits. At least five are needed.
  - Any comment that can help the instructor to understand your project.

Programming guideline (considered for grading):

- Your CSS code must pass the W3C CSS validator. Your JavaScript code should pass JSLint with no errors. Your .js file must run in JavaScript strict mode by putting "use strict"; at the top.
- Minimize global variables, avoid redundant code, and use parameters and return values properly. Make extra effort to minimize redundant code. Capture common operations as functions to keep code size and complexity from growing. You can reduce your code size by using the this keyword in your event handlers.
- Some global variables are allowed, but it is not appropriate to declare too many; values should be local as much as possible. If a particular value is used frequently throughout your code, declare it as a global "constant" variable named IN_UPPER_CASE and use the constant throughout your code. Do not store DOM element objects, such as those returned by the $ or $$ or document.getElementById functions, as global variables.
- Your JavaScript code should have adequate commenting. The top of your file should have a descriptive comment header describing the assignment, and each function and complex section of code should be documented.
- Format your code by properly using whitespace and indentation. Use good variable and method names. Avoid lines of code more than 100 characters wide.
- Separate content (HTML), presentation (CSS), and behavior (JS). Your JS code should use styles and classes from the CSS rather than manually setting each style property in the JS. For example, rather than setting the .style of a DOM object, instead, give it a className and put the styles for that class in your CSS file. Style properties related to x/y positions of tiles and their backgrounds are impractical to put in the CSS file, so those can be in your JS code.
- Use unobtrusive JavaScript so that no JavaScript code, onclick handlers, etc. are embedded into the HTML code.