Lab 8

Due Nov 14, 2016 by 5pm **Points** 100 **Submitting** a file upload

CS-546 Lab 8

Simple Browser APIs and jQuery

You will make a simple server that renders a single page. This page will keep track of several values using client side JavaScript; you will detail the keys you use and their respective values on this single page in a table with 2 columns.

The columns are:

1. The key name you used
2. The value you are storing

You will have a form with a single input on the page. It can take a number, select, textarea, whatever you desire.

The data you will detail in the table:

1. How many times a 1.5 second interval has occurred since the page has loaded; this will be cumulative across all the times the page has loaded. This means that if you load the page and stay on it 5 seconds before refreshing, you will have gone through 3 intervals. This should be stored in localStorage each update, as well as updated on the table. After refreshing, the timer will restart but you will keep accumulating onto that same total.
2. You will record how many times the form has been submitted. You will use jQuery for the event handling.
3. You will record what the last inputed value was. You will use jQuery for the event handling.
4. You will keep track of the **location hash** (https://developer.mozilla.org/en-US/docs/Web/API/WindowEventHandlers/onhashchange) and update a table column when the document is loaded or the hash changes. *The page should not have to be reloaded to trigger the update on the table*

You will use jQuery to do all the DOM Manipulation.

Your page must pass accessibility tests using **tota11y** (http://khan.github.io/tota11y/) .

References and Packages

Basic CSS info can easily be referenced in the **MDN CSS tutorial** (https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started) . If you need a quick CSS referfence,

You will use the **express-handlebars** package as your templating engine.

You can reference the **express-handlebars repository** (https://github.com/ericf/express-handlebars/) for details on adding the module; you may also want to check out the **handlebars website** (http://handlebarsjs.com/) for details.

You will use the **express** package as your server.

You can read up on **express** (http://expressjs.com/) on its home page. Specifically, you may find the **API Guide section on requests** (http://expressjs.com/en/4x/api.html#req) useful.

You may use the **lecture 4 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_4) as a guide.

You may use the **lecture 5 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_5) as a guide.

You may use the **lecture 6 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_6) as a guide.

You may use the **lecture 8 code** (https://github.com/Stevens-CS546/CS-546-WS-Summer-1/tree/master/Lecture%20Code/lecture_8) as a guide.

Requirements

1. You **must not submit** your node_modules folder
2. You **must remember** to save your dependencies to your package.json folder
3. You must do basic error checking in each function
    1. Check for arguments existing and of proper type.
    2. Throw if anything is out of bounds (ie, trying to perform an incalculable math operation or accessing data that does not exist)
    3. If a function should return a promise, instead of throwing you should return a rejected promise.
4. You **must remember** to update your package.json file to set `app.js` as your starting script!
5. **Your HTML must be valid** (https://validator.w3.org/#validate_by_input) or you will lose points on the assignment.
6. Your HTML must make semantical sense; usage of tags for the purpose of simply changing the style of elements (such as `i`, `b`, `font`, `center`, etc) will result in points being deducted; think in terms of content first, then style with your CSS.
7. **You can be as creative as you'd like to fulfill front-end requirements**; if an implementation is not explicitly stated, however you go about it is fine (provided the HTML is valid and semantical). Design is not a factor in this course.
8. **Your client side JavaScript must be in its own file and referenced from the HTML accordingly**.