ITMD 441/541

Lab 3 – Simple Tip Calculator

Introduction

This first lab will have you write a very simple HTML and CSS UI for a tip calculator and add the JavaScript needed to make it work.

Requirements

- Use the included wireframe mockup as a guide to what components should be in the app UI and a general layout.
- You need to include everything in the mockup, but you may lay it out differently. Just make sure it is clear to understand how it would work.
- The UI needs 4 text inputs, 1 range slider input, and the appropriate labels and text. Make sure you use actual <label> elements to label your form controls with either the explicit or implicit method. You need to wrap all form controls inside a form tag, but you do not need to set the method or action attributes, JavaScript will process it.
- The **Tip Percentage, Tip Amount,** and **Total Bill with Tip** boxes **should be disabled** so a user cannot type in any values. Style them differently to show this too. Lookup disabled attribute on a form control.
- The Bill Total field should be a text field the user can type a number in.
- The Tip field should use an input of range type with a min value of 0 and max value of 100. Lookup input type of range.
- The Tip Percentage, Tip Amount, and Total Bill with Tip fields should be disabled text inputs. The user should not be able to type values in these. You will put the values in these with JavaScript.
- The app needs to listen for an “input” or “change” event on the form element. Either will fire anytime any of the form’s controls are changed. Input fires continuously, change only fires after moving to another field.
- When you hear the event, you should read the value out of the Bill Total field and also the value in the tip input slider. Then you should write the value of the tip slider into the Tip Percentage field and also calculate the other two fields. Tip Amount should be the Tip Percentage of the Bill Total and Total Bill with Tip is the Bill Total field plus the Tip Amount field. Use the disabled inputs value property to write those numbers in them.
- You should be able to keep changing values in Bill Total or the Tip Slider and the form will keep updating.
- The values you put in the Tip Amount or Bill Total with Tip should only use 2 decimal places. Look up the toFixed() method on a number object.
- You should also do some validation to make sure if someone enters a non-number in the Bill Total field, they get some kind of notice or error message.
- When testing make sure all values make sense. For example, a bill of 100 with a tip of 10% should have a tip amount of 10 and a total with tip of 110.

- Make sure the form has some CSS styling, so it stands out and doesn't just blend into the background. Borders, background colors, or anything like that. You are free to style it how you want but you must apply enough CSS so that the HTML doesn't look like it would by default.
- Make sure the HTML document uses the HTML5 page skeleton we talked about in class.
- Add your CSS and JS as an **external files**.
- You may use a CSS framework if you like.
- 465 students just need to make it work and look nice on desktop. 565 students need to make sure it still is usable on smaller screen sizes like mobile. Resize your browser window or use the develop tools in mobile mode to test. 565 Students should use some media queries.
- Publish your finished work to your GitHub repository and GitHub Pages.
- Test the GitHub Pages URL in your browser to ensure it all works before you submit.
- Submit your both your GitHub Repository and your GitHub Pages hosted file URLs in the blackboard assignment text field.

Graduate Additional Requirements

If you are involved in **any section of 541** you need to complete the additional requirements listed here.

Make sure the user interface works well at mobile phone sized screens. You can do this with responsive CSS or with a creative way of laying out the UI, so it works well on both desktop and mobile. Use browser dev tools to test on mobile. Also, there is a higher expectation on the quality of CSS used for 541 students.

README File

Not Required for this Lab

Due Date / Late Policy

This assignment is due **Sunday October 29, 2023 11:59 PM Chicago Time**. **No Extensions or Exceptions. Late policy in syllabus applies.**

Submission Guidelines

You must upload your submission, to the blackboard assignment by the due date. The submission must be in the following format and structure. If you do not submit your assignment exactly as specified, you will receive an immediate 10% deduction.

Submission Format Specification:

Commit all files used to your main class GitHub Repository in a separate directory or a new repository just for this lab and submit to the Blackboard assignment the URLs to both your GitHub repository and the GitHub Pages hosted page.

Mockup