Lab 10: Client-side JavaScript

Due date: Oct 30, 2021, 4:00am

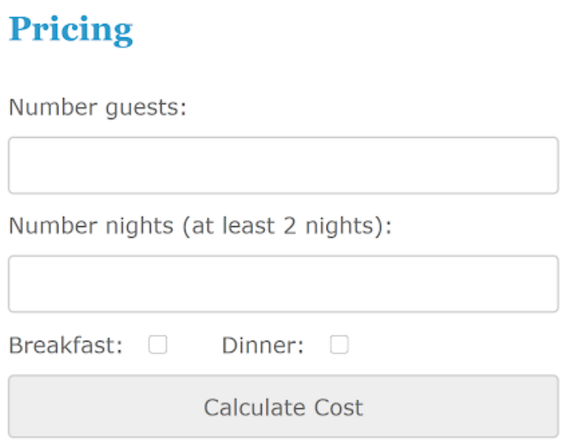
Overview

In this lab, you will implement client-side scripts to process a form and display feedback on a page.

The Starter Project

The starter project is a version of the pacific trails resort case study with a new page called pricing. The pricing page is supposed to give the site visitor a quote for a stay at pacific trails resort. In this lab, you will only work on the behavior of the pricing page.

The pricing page consist of a form with two <input> elements of type number, two <input> elements of type checkbox, and the "Calculate Cost" button, which is implemented by a <div> element:



If you open the file pricing.html in VS Code, you will find several <div> elements with class `display-none` that are not displayed on the page because their class `display-none` is styled to set the CSS property `display` to none.

- The first <div> element of class `display-none` wraps an entire table. The table should be used to display the quote for a stay at pacific trails resort for the submitted form input. You will implement the behavior that determines the value for the table columns and that displays the table upon form submission. If you submit the form in the starter project, nothing happens.
- The other two <div> elements of class `display-none` are located in the form and represent an error message for the first and second input field, respectively. If an invalid value is entered in the form, the corresponding error message should be displayed. You will implement the validation and display of the error messages in case of invalid input.

Instructions

In this lab, you add all code to the JavaScript file `js/pricing.js`. Do not make any changes to the stylesheet or to the HTML files.

Part 1: Validation of number guests

1. Define a function that checks whether the input field for the number of guests contains a positive value, i.e. a value greater than 0:

    a. Retrieve the value of the input field for the number of guests. For example:

    ```
    let numGuests = document.getElementById('num-guests').value;
    ```

    b. If the input field is empty or contains a non-positive value, i.e. a value less than or equal to 0, remove the class `display-none` from the <div> element with id `guests-error` and return `false`.

    Note that the variable `numGuests` as defined above is of type string. To obtain the number represented by the string apply the method `Number.parseInt(numGuests)`. To remove the class from the <div> element retrieve first the <div> element:

    ```
    let guestsError = document.getElementById('guests-error');
    ```

    Then remove the class:

    ```
    guestsError.classList.remove('display-none');
    ```

    c. If the input field contains a positive value, add the class `display-none` to the <div> element with id `guests-error` and return `true`.

    You can add the class `display-none` to the element represented by variable `guestsError` as follows:

    ```
    guestsError.classList.add('display-none');
    ```

2. Register the function as event handler with the <input> element with id `num-guests` for the focusout event. I suggest placing the instruction(s) to register any event listeners at the top of the JavaScript file. Note that these instructions are not wrapped in any function or method.

3. Test the script:

    a. Enter a value ≤ 0 in the input field for the number of guests and then move the cursor out of the input field. The error message should be displayed below the input field.

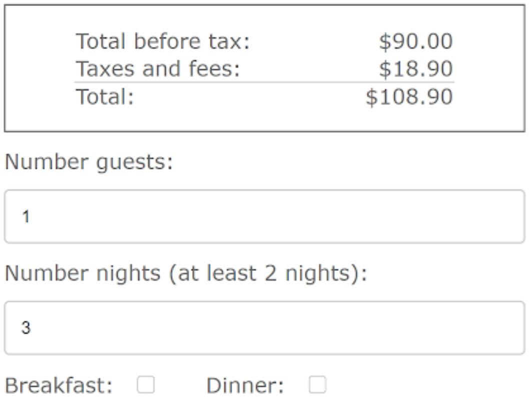
    b. Enter a a positive value in the input field and move the cursor out of the input field. The error message should disappear.

Part 2: Validation of number nights

Analogously to Part 1, validate the input field for the number of nights. Note that a valid input value must be greater than 1. Test that the error message is displayed for an invalid value as soon as the input field for the number of nights looses focus. Test that the error message disappears if the value is corrected and the input field looses focus.

Part 3: Calculate and display cost

1. Implement a function that calculates and displays the cost for the entered user input:
   a. Check if the input for the number of guests and nights is valid. Use the validation functions implemented for Part 1 and 2. The return value of these functions will indicate whether the input is valid.
   b. If one or both fields are invalid, add the class `display-none` to the <div> element with id `price-result` and terminate the function.
   c. If both fields are valid,
      i. calculate the cost before taxes: The cost is $30 per person per night. If breakfast is included, $10 is added per person and per night. If dinner is included, $20 is added per person per night.
      ii. calculate the taxes and fees: The taxes and fees are 21% of the cost.
      iii. calculate the total: The total is the sum of the cost and the taxes & fees. (See also the examples in the images below.)
      iv. Set the inner HTML of the suitable <td> elements in the HTML table in file pricing.html to the cost, the taxes, and the total, respectively.
      v. Remove the class `display-none` from the <div> element with id `price-result`.
2. If the above function is getting too large, say more than 40 lines, refactor the function by writing one or more helper functions. For example, a helper function may calculate the cost values.
3. Register the function as event handler with the Calculate-Cost button, i.e. register the function with the <div> element with id `price-button`, for the click event.
4. Test the script:
   a. Enter valid values, click the Calculate Cost button, and check that the correct quote that is displayed. (See also the examples in the images below.)
   b. Enter different valid values, click the Calculate Cost button, and check that the quote is correctly updated.
   c. Enter an invalid value, click the Calculate Cost button, and check that no quote is displayed.
5. If you have not yet done so, format the displayed cost values as shown in the four examples on the next page.

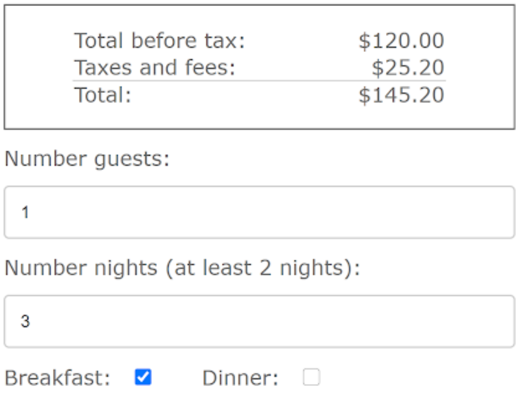


Total before tax:	$90.00
Taxes and fees:	$18.90
Total:	$108.90

Number guests: 1

Number nights (at least 2 nights): 3

Breakfast: □ Dinner: □

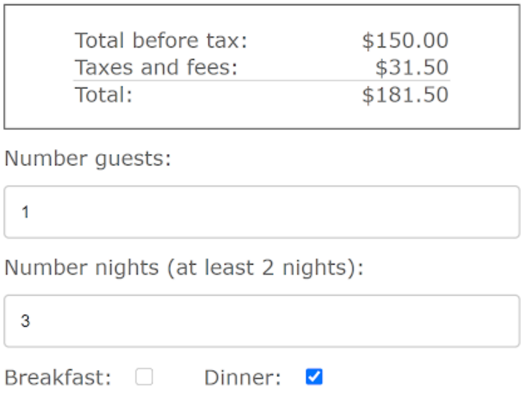


Total before tax:	$120.00
Taxes and fees:	$25.20
Total:	$145.20

Number guests: 1

Number nights (at least 2 nights): 3

Breakfast: ☑ Dinner: □

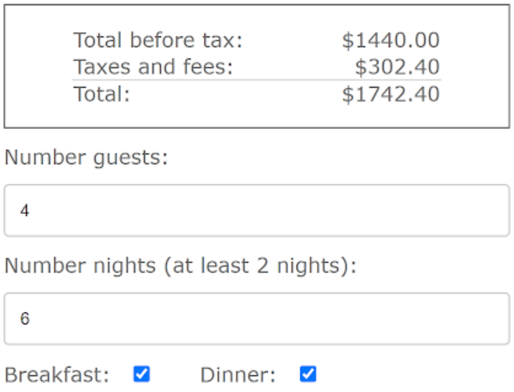


Total before tax:	$150.00
Taxes and fees:	$31.50
Total:	$181.50

Number guests: 1

Number nights (at least 2 nights): 3

Breakfast: □ Dinner: ☑



Total before tax:	$1440.00
Taxes and fees:	$302.40
Total:	$1742.40

Number guests: 4

Number nights (at least 2 nights): 6

Breakfast: ☑ Dinner: ☑

Submission

Archive the folder that contains all files with your solution using the ZIP format and submit your solution on the course website before the due date. Make sure that the name of the submitted zip-file contains your name.

Grading

Part 1: 2 points

Part 2: 2 points

Part 3 Steps 1-4: 5 points

Part 3 Step 5: 1 point