

Case Study 04(i) – Form Validation



Job validation exercise steps:

1. **Create job.html:** set the form to send data to a PHP script (show_get.php or show_post.php)
2. **Add JavaScript Form Validation:** Link or embed a formvalidation.js script in job.html.
3. **Implement JavaScript Validation:** Register event: use addEventListener to trigger validation on form submission; Create Validation functions to validate input fields using specific rules.
4. **Handle Errors:** If input is invalid, display an alert with clear error message.

JavaJam Coffee House

Jobs at JavaJam

Want to work at JavaJam? Fill out the form below to start your application. Required fields are marked with an asterisk *

*Name:

*E-mail:

Start Date:

*Experience:

Copyright © 2014 JavaJam Coffee House
yourfirstname@yourlastname.com

Case Study 04(i) – Form Validation Criterias



Job Validation Criteria:

1. The name field must contain alphabet characters and character space.
2. The email field contains a username part follows by "@" and a domain name part. The username contains word characters including hyphen ("-") and period (".").
3. The domain name contains two to four address extensions. Each extension is string of word characters and separated from the others by a period ("."); The last extension must have two or three characters.
4. The start date cannot be from today and past.
5. The experience field cannot be empty.

Notes: You are encouraged to create additional validations, but the above-mentioned validations are mandatory. Additional credit will be awarded for defining extra validations.

Case Study 4(ii) – Menu Update



Menu Update exercise steps:

1. **Page Layout:** Create six text boxes in your page layout to update the quantities and total prices for the coffees. **Radio Buttons:** For each type of coffee, include radio buttons to let customer select whether they want a single or double shot. The price will vary depending on their selection. **Unique Naming:** Assign a unique name to each text box. This is crucial so that after you perform calculations in your JS file, you can correctly update the subtotal and total prices on the HTML page.
2. **JavaScript Function:** Create a function in your JS file to calculate the subtotal and total price based on the customer's selections and input.

Case Study 4(ii) – Menu Update

- Modify the “Menu” page to include order quantity and options for single or double shots. The sub-total for each order should be displayed along the total amount below the sub-totals. The sub-totals and total should be computed using JavaScript function..

The screenshot shows the JavaJam Coffee House menu page. The menu is titled "Coffee at JavaJam" and lists three coffee options: "Just Java", "Cafe au Lait", and "Iced Cappuccino". Each option has a description and a price for a single shot. The "Total price" is displayed at the bottom right. Annotations with arrows point to specific areas for updates:















- Add column of text boxes for quantity inputs (points to the right side of the menu items)
- Add column of text boxes for sub-total prices (points to the right side of the menu items)
- Add text box for total price (points to the "Total price" field)
- Add radio buttons for price selection (points to the "Single" and "Double" options for each coffee)

Coffee	Description	Single	Double
Just Java	Regular house blend, decaffeinated coffee, or flavor of the day. Endless Cup \$2.00		
Cafe au Lait	House blended coffee infused into a smooth, steamed milk. Single \$2.00 Double \$3.00		
Iced Cappuccino	Sweetened espresso blended with icy-cold milk and served in a chilled glass. Single \$4.75 Double \$5.75		

Total price

Casestudy 04(ii)



-  bullet
-  chinese
-  index
-  JavaJam
-  jobs
-  jobvalidation
-  menu
-  MenuUpdate
-  music
-  music_1
-  music_2
-  show_get
-  title
-  windingroad

Menu.html: create a form and receive the user input for each of the text areas, for each radio button and text area, you will need to use different names, this is to calculate the subtotal based on the customer order.

MenuUpdate.js: after receiving the user's input, we create functions to calculate the subtotal and total price on the orders



Event Registration for Cafe

```
var CafeNode1 = document.getElementById("Qty_Cafe_Single");  
var CafeNode2 = document.getElementById("Qty_Cafe_Double");  
var CafeNode3 = document.getElementById("QtyCafe");
```

Add event listener

```
CafeNode1.addEventListener("change", Cal_Cafe, false);  
CafeNode2.addEventListener("change", Cal_Cafe, false);  
CafeNode3.addEventListener("change", Cal_Cafe, false);
```

CafeNode1: store the price of single shot cafe

CafeNode2: store the price of double shot cafe

**CafeNode3: store the quantity of the selection,
(single or double)**



MenuUpdate.js

```
function Cal_Cafe(event)
{
```

```
    var qtyCafechk = event.currentTarget;
```

```
    var pos = qtyCafechk.value.search(/^[0-9]*$/);
```

```
    if(pos != 0)
```

```
    {
```

```
        alert("You entered (" + qtyCafechk.value + ") is not correct. \n" +  
            "Only NUMBER are allowed.");
```

```
        qtyCafechk.value = "";
```

```
        document.getElementById("Price_Cafe").value = "";
```

```
        qtyCafechk.focus();
```

```
        return false;
```

```
    }
```

Find element with ID Price_Cafe in your menu.html, and update the value

```
    if(CafeNode1.checked == true)
```

```
    {
```

```
        Total_Cafe_Price = CafeNode1.value * CafeNode3.value;
```

```
        document.getElementById("Price_Cafe").value = parseFloat(Total_Cafe_Price).toFixed(2);
```

```
    }
```

Calculate sub total price

```
    if(CafeNode2.checked == true)
```

```
    {
```

```
        Total_Cafe_Price = CafeNode2.value * CafeNode3.value;
```

```
        document.getElementById("Price_Cafe").value = parseFloat(Total_Cafe_Price).toFixed(2);
```

```
    }
```

```
Cal_total();
```

Calculate total price

Case Study 04 Additional Approach



- **Base Version (85%):**
Traditional approach (HTML, CSS, JS, PHP)

- **Additional approach to create or improve case study 04 (15%):**

Modern Approaches for Job Validation and Menu Update: React.js with states and hooks. Use `useState()` and `useEffect()` for real-time form validation; For the MenuUpdate, use `useState()` to track selected menu items and quantity. Use `useEffect()` to automatically update subtotal when items change. Other options: Vue.js, Angular, jQuery.

**CSS Frameworks to improve styling:
Bootstrap, Tailwind CSS, and etc.**



Thanks



Lecturer: Dr. Hu Xiao

Email: xiao.hu@ntu.edu.sg