

Bo Sheng Lee

COMS 319

Professor Ali

7 September 2020

## Report

### **Solution Approach**

First, I have to create two .html file, called validation1.html and validation2.html, and two.js file called validation1.js and validation2.js. Then I have to create a form for each .html file. Other than that, I have to also include validation 1.js in validation1.html and validation2.js in validation2.html. Then I start to add different input in the form of validation1.html. For example, first name, last name, gender, and state. The first name and last name are text input by the user, and gender and state are select input by the user. To validate the first name and last name, I use the .match function to compare user input with a format of string  `/^[0-9a-zA-Z ]+$/` , which means only allow input from 0-9,a-z, A-Z, and space. To validate the gender and state, I simply check whether the user selects an empty option or a valid option. To begin the validation and direct to validation2.html, I create a “Continue” button.

Then, for validation2.html, I add email, phone, and address input in the form. To validate email, I first split the string into two with respect to the character “@.” Then I check

whether the size array is equal to two. After that, I check the first element string in the array to make sure no special character in the string. Then I split the second element string in the array with the character “.”. Again, I check the split string to make sure no special character in it. To validate phone input, I first split the input string into an array with the character “-.”. If the user inputs the phone number with format xxxxxxxxxx, I will only have an array with size 1. Then I check each element in the array to make sure only numeric character is in the string. To validate the address, I first split the input string into two with the character “,”. If the array size is less than 2 means, the input is incorrect. Then I check the second element of the array to make sure only 2 characters in the string and all of the characters are capital letters. To begin the validation, I add a “Submit” button.

There is a true or false image display that depends on the user’s input for every input. The function getImage() will take the boolean variable to decide which image to return. The boolean variable is returned by every validateFuntion() in .js file. Then the image will be appended to the input textbox. Other than the image, an error message will also display when the user has the incorrect input. The getNotifications() function will return an error message depends on the ID parameter. Then the error message will also append to the input textbox.

## Screenshots

Screenshots 1: Example of correct input where the first name and last name accept alphanumeric characters.

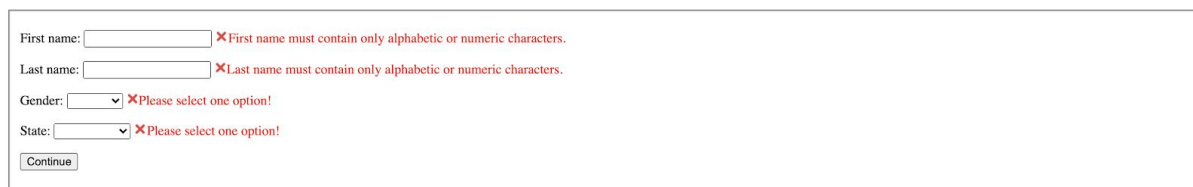
### Validation Form



A screenshot of a validation form titled "Validation Form". It contains four input fields: "First name:" with the value "Bo Sheng1", "Last name:" with the value "Lee2", "Gender:" with a dropdown menu showing "Male", and "State:" with a dropdown menu showing "California". Each field has a green checkmark to its right. At the bottom left is a "Continue" button.

Screenshot 2: Example of empty input by user

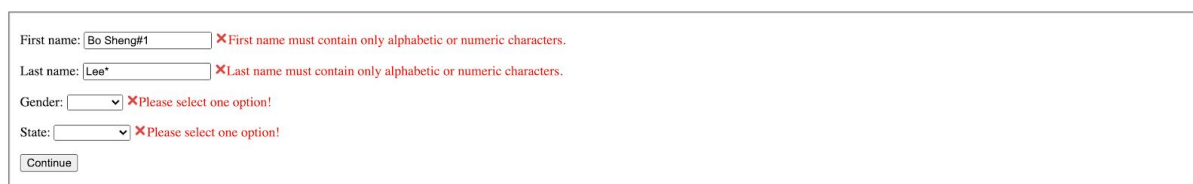
### Validation Form



A screenshot of a validation form titled "Validation Form". It contains four input fields: "First name:", "Last name:", "Gender:", and "State:". Each field is empty and has a red error message to its right: "✖First name must contain only alphabetic or numeric characters.", "✖Last name must contain only alphabetic or numeric characters.", "✖Please select one option!", and "✖Please select one option!". At the bottom left is a "Continue" button.

Screenshot 3: Example of incorrect input by the user with characters other than alphanumeric characters.

### Validation Form



A screenshot of a validation form titled "Validation Form". It contains four input fields: "First name:" with the value "Bo Sheng#1", "Last name:" with the value "Lee\*", "Gender:" with a dropdown menu showing an empty selection, and "State:" with a dropdown menu showing an empty selection. Each field has a red error message to its right: "✖First name must contain only alphabetic or numeric characters.", "✖Last name must contain only alphabetic or numeric characters.", "✖Please select one option!", and "✖Please select one option!". At the bottom left is a "Continue" button.

Screenshot 4: Example of correct input by the user with only alphanumeric characters email, 10 numbers in phone and city, STATE in the address section.

## Contact Information

Email:	<input type="text" value="leebosheng717@gmail.c"/>	✓
Phone:	<input type="text" value="5154231155"/>	✓
Address:	<input type="text" value="Ames,IA"/>	✓
<input type="button" value="Submit"/>		

Screenshot 5: Example of empty input by the user.

## Contact Information

Email:	<input type="text"/>	✗Email should be in form xxx@xxx.xxx, which x should be alphanumeric!
Phone:	<input type="text"/>	✗Phone should be in the form xxx-xxx-xxxx or xxxxxxxxxxx, which x should be numeric
Address:	<input type="text"/>	✗Address should be in the form of city.STATE.
<input type="button" value="Submit"/>		

Screenshot 6: Example of the phone section accept formate xxx-xxx-xxxx.

## Contact Information

Email:	<input type="text" value="leebosheng717@gmail.c"/>	✓
Phone:	<input type="text" value="515-423-1155"/>	✓
Address:	<input type="text" value="Ames,IA"/>	✓
<input type="button" value="Submit"/>		

Screenshot 7: Example of incorrect input by the user where input with special character in the email section, more than 10 numbers in the phone section, and state with more than 2 alphabets in the address section.

## Contact Information

Email:	<input type="text" value="leebosheng\$%^@gmail."/>	✗Email should be in form xxx@xxx.xxx, which x should be alphanumeric!
Phone:	<input type="text" value="12345678901"/>	✗Phone should be in the form xxx-xxx-xxxx or xxxxxxxxxxx, which x should be numeric
Address:	<input type="text" value="Ames,iowa"/>	✗Address should be in the form of city.STATE.
<input type="button" value="Submit"/>		