Wk2: Assignment: Form Validation (3h)

Form Validation using Regular Expression

You will create JavaScript code using Regular Expressions, to validate the data in each form field. The data being validated will need to meet a predefined format. If the format is NOT met for a field, the input field will become RED, and an error message will display. If the format is met, the input field will become GREEN.

Objectives & Outcomes

Successful completion of this activity will show that you can:

- Understand the details of "Regular Expressions" (how it works, when to use it, understand the formatting).
- Exhibit working knowledge of "Regular Expressions", by writing JavaScript to validate form fields for a generic username, email, phone number, social security number, and password.
- Have the knowledge to analyze data that regular expressions can be applied to, and know that the regular expression written meets all formatting requirements.

Level of Effort

This activity should take approximately 180m to complete. It will require:

- 0m Research
- 15m Prep & Delivery
- 165m Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

Reading & Resources

ALL Rubrics *(necessary)*
This link has all the files you need to get started with this programming assignment.

Assignment Files *(necessary)*
This link has all the files you need to get started with this programming assignment.

Instructions

Getting Started:

- In your local branch of the PWA1 GIT repo, you will need to create a directory entitled "goal5_assign_validation".
- Download the .zip file from FSO for this assignment and unzip it in the directory above, "goal5_assign_validation". This is where all your assignment development files will reside.
- Using JavaScript comments, place your name, date and assignment at the top of the JavaScript file.
- Make sure you adhere to proper folder constructs, if applicable (css, images, js, etc)
  (Online) Please watch the associated screencast that shows the finished assignment and explains the rubric and requirements

Criteria:

To obtain full credit on the assignment, the following criteria must be adhered to as well as satisfying all items on this assignment's rubric.

Using regular expressions, you will need to create JavaScript code to validate each of the different formatted fields. Each field has a very specific format which is outlined below.

1. An HTML form containing fields has been provided to you in the index.html file. The fields on the form are as follows:

- username (id = f_username)
- email (id = f_email)
- phone number (id = f_phone)
- social security number (id = f_ssn)
- password (id = f_password)

The id's have been provided to you and is located in the HTML file for each input field.

Each field will need to meet the following formats:

- **username:** Has a capitalized first character for the firstName and lastName. (i.e "John Doe", "Mary Ann Doe)
- **email:** Matches a basic email address, and checks that they are of the proper form. Check to ensure the top level domain is between 2 and 4 characters long, but does not need to check the specific domain against a list (especially since there are so many of them now). (i.e. “account@domain.topLevelDomain”)
    1. **account**: This can be any number of alpha-numeric characters, must start with an alphabet.
    2. **@**: this at-sign must exist in an email address
    3. **domain**: This can be any number of alpha-numeric characters, must start with an alphabet.
    4. **. (the dot)** : This is a required special character.
    5. **topLevelDomain**: The top level domain will between 2 and 4 alpha characters long, this is required in the email address name.
- **phone number**: The phone number must meet this format (###)###-####.
- **social security number**: The social security number must meet this format ###-##-####.
- **password** : The password's first character must be a letter, it must contain at least 4 characters and no more than 15 characters and no characters other than letters, numbers and the underscore may be used

2. A skeleton of the program is being provided to you in the validate.js file.

3. You will create an onsubmit function that will contain a call to the function validateField. An argument needs to be passed in the validateField function call. The argument will be the ID name of the input field. You must dynamically retrieve the ID name from the DOM/HTML. You will need one call to the validateField function for each input field that needs to be validated.

4. A skeleton of the validateField function has been provided to you in the validate.js file.

5. In the validateField function, you will need to create an else-if statement for each input field id. See the IF statement example format in the validate.js file.

6. In the validate.js file you will see a variable named "pass". The string value will need to be replaced with JavaScript code that will test the pattern against the input value.

7. All the code to make the background red, green, or white has been provided. All you need to do is generate the code above.

Deliverables

You will submit your project via GIT. This is where your work will be graded. You will need to ensure you have at least 6 resonable commits.