CS618 Lab7 Web Application development

Please name the file as "your roll_no"-HW7.html and submit the HTML file with your Javascript code in the github repo. Make a new branch for lab 7.

Create a simple HTML page with a form where one book a round trip airline ticket. You don't (obviously) have to worry about the actual booking part. The webpage should have the following fields and do the following: The form should have the following fields (in the body part):

1. A textbox for each of
    - "First Name"
    - "Last Name"
    - "Last Name"
    - "Email Address"
    - "Origin city"
    - "Destination city"
2. "Departure date" -- in terms of year, month and day. You can use either a textbox for the entire date, or a dropdown menu for each of the year, month, day.
3. "Return date" -- in terms of year, month and day. You can use either a textbox for the entire date, or a dropdown menu for each of the year, month, day.
4. A textarea for "General comments".
5. A submit button.
6. A reset buton.

All the fields in the form except for "General comments" are mandatory. That is, the user should provide valid non-null values for all of them. Write Javascript functions given below to verify the following:

1. First name and last name are valid. They should have non-null, at least one valid (non whitespace) character.
2. Check that the email address is not null and has a format "a@b.c" or "a@b.c.d". **NOTE:** For this HW, we will use only the above two email address formats and nothing else. Here "a", "b", "c" and "d" should be non null valid values with at least a single character. If any of these ("a", "b", "c" or "d") has only one character, it should be in the range 'a-z' or 'A-Z'.
3. Origin city and destination city should be valid. They should be non-null and should have at least one valid character. Origin city should be different from the destination city.
4. Date of departure and date of arrival are both valid. E.g., Jun 31, Feb 29 2005, etc. are invalid. Make sure your program takes care of Feb 29 in a leap year.
5. Date of departure is sometime in the future.
6. Date of arrival is later than the date of departure.
7. "onSubmitFn( )" that is executed if the "Submit" button is pressed. The function should verify all the input, using the other functions above.

If any of the above conditions are not met, a popup box should come up and give an appropriate error condition, pointing out what went wrong. If everything is fine, take to another page that confirms that the application is received.

8. "onResetFn( )" that should clear the values of the fields of the form.

Note:

1. For any form with the name "myForm", you can retrieve the value of a text field in the form using "document.myForm.myTextField.value", where myTextField is the name of the text field.
2. For a "date" field (departure date and arrival date), you can use one of two formats. In the first format, you can have separate values for day, month and year. In the second format, you can use a string format, something like "2018-09-20".
3. You will find the "Date" object of Javascript useful. You can find details here and here. In particular, you will find the "getTime()" method useful to compare if one date comes before or after another date.
4. The code segment

```
var dateFormat = /^\d{4}[\-](0?[1-9]|1[012])[\-](0?[1-9]|[12][0-9]|3[01])$/;
var flag = dateString.match(dateFormat);
```

can be useful to match a string to a date format "YYYY-MM-DD". That is the variable "flag" will be null if any string (dateString) does not match the above format, non-null if it matches.

TASK 2:

In this homework, you will write a simple calculator as follows. Create a form that will hold the entire calculator with the following fields.

1. An input text that is used to hold the value.
2. Ten buttons, the first one for '0', the second for '1', the third for '2' and so on, with the last being '9'. When any of these buttions is pressed, the value in the input text should be updated.
3. Four more buttons '+', '-', '*' and '/' for addition, subtraction, multiplication and division. These should be used for the addtion, subtraction, multiplication and division operations, respectively.
4. A button "." that is used to create a decimal value.
   Note: The user should be able to use a button only once for any number. That is, something "3.4.5" is an invalid number and should not be allowed.
5. A button "+/-" that is supposed to toggle the value of the current number from positive to negative and negative to positive.
6. A button '=' that should show the updated value (result) in the input text.
   Note: You can use *eval* function on the input text value to find the result
7. A button called 'C' that is used to clear the value in the input text.

Note:

1. You can add two numbers (e.g., 25.34 + 30.12) at a time or multiple numbers (25.34 + 30.12 + 11.9 + 36 + ..) at a time.
2. If you are adding multiple numbers at a time, you need to show all the numbers in the input box (25.34 + 30.12 + 11.9 + 36 + ..) and only the result after the "=" is pressed.
3. If you are adding two numbers at a time, you can show both the numbers (e.g. 25.34 + 20.12) on the screen or one at a time (e.g., show 25.34 first, then show only 30.12 after the "+" symbols is pressed). Show the result after the "=" is shown.
4. You can store the numbers in a local variable or evaluate them from the text input field using the "eval" function.