CSE 216 – Programming Abstractions (Spring 2019)

Programming Assignment # 5

In this assignment, you will create JavaScript/HTML based webpages with certain considerations. The assignment has two questions together comprising 50 points.

Question I. Abstract submission form **(35 pts)**

Create a HTML form that accepts abstract for submission to a particular conference. The form has fields and associated conditions as shown in the following table. Create custom JavaScript functions which will check each field in the form to check whether it satisfies a particular condition. On submission of a form, if the fields do not satisfy a particular condition, an alert is shown to the user along with highlighting the particular fields in the form. In case the form is correctly filled, on submission the details of submission are shown in the same page at the bottom of the form.

Form field	Type	Value	Restrictions	Points
Title of the form	Display text	Abstract Submission Form	Show in bold	1
Abstract title	Input text	To be entered by user. E.g. **A framework for the comparison of mobile patient monitoring systems**	1. Should contain only alphanumeric characters and space 2. Maximum length = 70 characters	3
Authors names	Input text	To be entered by user E.g. **Pravin Pawar*, Val Jones, Bert-Jan F Van Beijnum, Hermie Hermens**	1. Should contain only alphabetical characters, space and 3 special characters '-', ',' and '*'. 2. Maximum number of authors is 5, where author names are separated by comma. 3. There must be a * after the name of corresponding author. Only one corresponding author is possible.	3
Corresponding Author's Affiliation	Input text	To be entered by user. E.g. **Department of Telemedicine, University of Twente, Enschede, The Netherlands.**	1. Should contain only alphabetical characters, 2 special characters '-', ',' and space. 2. Maximum length = 100 characters.	3

Corresponding author's email	Input text	To be entered by user. E.g. **pravin.pawar@utwente.nl**	See the description following this table.	3
Corresponding author's phone number	Input text	To be entered by the user. E.g. **+31 – 53 489 3265**	1. Must start with a + sign followed by valid country code 2. Should contain only digits and following special characters: '+', '-' and space. The '+' symbol must only occur in the beginning. 3. Total number of digits should not be more than 14.	3
Abstract	Input text	To be entered by the user.	1. May contain alphanumeric as well as all special characters. 2. The maximum number of words should be 200. 3. The maximum number of characters should not exceed 1500.	3
Key words	Input text	To be entered by the user. E.g. **patient monitoring, mobile computing, body area network**	1. Should contain only alphabetical characters, space and 3 special characters '-', ',' and space. 2. Individual keywords must be separated by comma. 3. At most 5 key words are allowed. 4. The total length of key words should not exceed 100.	3
Submit button	Input button	Submit	1. On clicking the Submit button, your JavaScript should get data from all the form fields and check with respect to above conditions. In case of any problems with the fields, alerts should be shown that contains list of all possible problems. 2. The corresponding fields with error values should be highlighted. 3. In case there are no errors, below the form the	Alerts: 3 Highlight fields with errors: 3 Display abstract: 4

			entire contents of the abstract should be displayed in a tabular format.	
Clear button	Clear button	Clear	1. On clicking this button, all the data in the form should be cleared.	3

Email address specification:

The format of e-mail address is: local-part@domain-part (max. 64@255 characters, no more 256 in total).
The local-part and domain-part could have different set of permitted characters, but that's not all, as there are more rules to it. In general, the local part can have these ASCII characters:

- lowercase Latin letters: abcdefghijklmnopqrstuvwxyz,
- uppercase Latin letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ,
- digits: 0123456789,
- special characters: !#$%&'*+-/=?^_`{|}~,
- dot: . (not first or last character or repeated unless quoted),
- space punctuations such as: "(),:;<>@[\] (with some restrictions),
- comments: () (are allowed within parentheses, e.g. (comment)john.smith@example.com).

Domain part:

- lowercase Latin letters: abcdefghijklmnopqrstuvwxyz,
- uppercase Latin letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ,
- digits: 0123456789,
- hyphen: - (not first or last character),
- can contain IP address surrounded by square brackets: jsmith@[192.168.2.1] or jsmith@[IPv6:2001:db8::1].

These e-mail addresses are valid:

- prettyandsimple@example.com
- very.common@example.com
- disposable.style.email.with+symbol@example.com
- other.email-with-dash@example.com
- x@example.com (one-letter local part)
- "much.more unusual"@example.com
- "very.unusual.@.unusual.com"@example.com
- "very.(),:;<>[]\".VERY.\"very@\ \"very\".unusual"@strange.example.com
- example-indeed@strange-example.com
- admin@mailserver1 (local domain name with no top-level domain)
- #!$%&'*+-/=?^_`{}|~@example.org
- "()<>[]:,;@\\"!#$%&'-/=?^_`{}| ~.a"@example.org
- " "@example.org (space between the quotes)
- example@localhost (sent from localhost)

- example@s.solutions
- user@com
- user@localserver
- user@[IPv6:2001:db8::1]

And these examples of invalid:

- Abc.example.com (no @ character)
- A@b@c@example.com (only one @ is allowed outside quotation marks)
- a"b(c)d,e:f;gi[j\k]l@example.com (none of the special characters in this local part are allowed outside quotation marks)
- just"not"right@example.com (quoted strings must be dot separated or the only element making up the local part)
- this is"not\allowed@example.com (spaces, quotes, and backslashes may only exist when within quoted strings and preceded by a backslash)
- this\ still\"not\allowed@example.com (even if escaped (preceded by a backslash), spaces, quotes, and backslashes must still be contained by quotes)
- john..doe@example.com (double dot before @); (with caveat: Gmail lets this through)
- john.doe@example..com (double dot after @)
- a valid address with a leading space
- a valid address with a trailing space

Submission:

Write your code along with necessary HTML and JavaScript in the `abstractsubmission.html` file.

Question 2. Read, display and summarize JSON data (15 pts)

A. Create a HTML page with necessary JavaScript that reads json data stored in a local file and displays this data in a tabular format in a webpage. The JSON file is provided together with this assignment. The name of JSON file is `life-expectancy-data.json`. The JSON file has information pertaining to 5 columns - "Income", "Life Expectancy", "Population", "Country", "Year". In total, there are 1540 total records. The first few rows are shown here: (7 pts)

Income	Life Expectancy	Population	Country	Year
815	34.05	351014	Australia	1800
1314	39	645526	Canada	1800
985	32	321675013	China	1800
864	32.2	345043	Cuba	1800
1244	36.5731262	977662	Finland	1800

B. Following the main table, create a summary table that shows summary statistics for each country. The summary statistics should be calculated by a JavaScript function. There are in total 19 countries. The summary statistics should show the mean values of following parameters for each country:

"Income", "Life Expectancy" and "Population". The mean values for some of the countries are shown below: (8 pts)

Country	Mean Income	Mean Life Expectancy	Mean Population
Australia	21989.92	69.47	13071980.46
Canada	22812.98	70.03	21336515.79
China	2498.12	56.45	886011894.50

Submission:

Write your code along with necessary HTML and JavaScript in the `jsontable.html` file.

Some Tips:

JavaScript email validation example:

https://www.w3resource.com/javascript/form/email-validation.php

To read a local json file, the second solution proposed in the following thread worked for me:

https://stackoverflow.com/questions/19706046/how-to-read-an-external-local-json-file-in-javascript/45035939

Populate json data to html table using Javascript:

https://www.encodedna.com/javascript/populate-json-data-to-html-table-using-javascript.htm

How to display results in the same page using JavaScript:

https://www.quora.com/How-do-I-display-results-on-the-same-page-in-JavaScript

Assignment submission: Submit only following files (no zip files) along with necessary Javascripts:

- `abstractsubmission.html`
- `jsontable.html`
- `life-expectancy-data.js/json`

Assignment Evaluation: A slot will be announced for evaluating the assignments. During this time the instructor/TAs will download and run your program and it will be checked for correctness using a few use-cases. If required, instructor/TA will contact you for clarifications.

Note:

If your code does not compile, it will not be graded.

Late submissions will not be accepted under any circumstances.

To be safe, always, ALWAYS, prepare to submit ahead of time, not exactly AT last moment!

Submission deadline: Monday 03 June, 11:59 PM