```
1
       /* jshint esversion: 8 */
 2
       // helper functions
 3
 4
 5
        * Function to add class
        * @param {*} className
 6
 7
        * @param {*} targetNode
 8
 9
       function addClass(className, targetNode) {
10
         targetNode.classList.add(className);
11
12
13
        * Function to remove class
14
        * @param {*} className
15
        * @param {*} targetNode
16
17
18
       function removeClass(className, targetNode) {
19
20
         targetNode.classList.remove(className);
21
22
       /**
23
        * Function to display Error after validation has been failed
24
        * makes div with help message visible and in red,
25
        * input's border is red and red icon
26
        * with exclamation mark is displayed in input field
27
        * @param {*} targetNodeInput
28
        * @param {*} targetNodeHelp
29
30
        */
31
       function displayErrorValidation(targetNodeInput, targetNodeHelp) {
32
         // input nodes
         addClass("is-invalid", targetNodeInput);
removeClass("is-valid", targetNodeInput);
33
34
35
         // help nodes
         removeClass("d-none", targetNodeHelp);
36
         addClass("invalid-feedback", targetNodeHelp);
removeClass("valid-feedback", targetNodeHelp);
37
38
       }
39
40
41
42
43
        * Function to display Error after validation has been failed
        * makes div with help message visible and in red, input's border is red
44
45
        * and red icon with exclamation mark is displayed in input field
46
        * @param {*} targetNodeInput
        * @param {*} targetNodeHelp
47
        */
48
       function displayPassedValidation(targetNodeInput, targetNodeHelp) {
49
50
         // input nodes
         addClass("is-valid", targetNodeInput);
51
         removeClass("is-invalid", targetNodeInput);
52
53
         // help nodes
         removeClass("d-none", targetNodeHelp);
54
55
         addClass("valid-feedback", targetNodeHelp);
56
         removeClass("invalid-feedback", targetNodeHelp);
57
       }
58
59
        * function to validate if input is NOT empty
60
        * @returns true or false
61
62
63
       function isNotEmpty(inputNode, helpNode) {
64
65
         if (inputNode.value === "") {
66
           helpNode.innerHTML = "This field is required";
67
           displayErrorValidation(inputNode, helpNode);
68
           return (false);
69
70
         } else {
```



version 2.13.4 There are 10 function (https://github.com/jshint/j: Function with the larg

About median is 2.

Documentation (/docs) has 2.5.

2.5 < llater

The most complex fu

Blog (/blog)

```
71
           helpNode.innerHTML = "All ok";
72
           displayPassedValidation(inputNode, helpNode);
73
           return (true);
74
         }
75
       }
76
77
        * function to validate if both arguments are met: regex and not empty
78
79
ลล
       function validateInput(inputNode, helpNode, regex, errorMessage) {
81
82
         if (inputNode.value === "") {
           helpNode.innerHTML = "This field is required";
83
           displayErrorValidation(inputNode, helpNode);
84
85
           return false;
         } else if (regex.test(inputNode.value) === false) {
86
           helpNode.innerHTML = errorMessage;
87
88
           displayErrorValidation(inputNode, helpNode);
89
           return (false);
90
         } else {
           helpNode.innerHTML = "All ok";
91
92
           displayPassedValidation(inputNode, helpNode);
93
           return true;
94
         }
95
       }
96
97
       //
            variables fetching the elements from the form
98
99
       const contactForm = document.getElementsByTagName("FORM")[0];
100
       // id strings for input
101
       const stringIdContactName = "contact-name";
102
103
       const stringIdEmail = "email";
104
       const stringIdTelephone = "telephone";
       const stringIdCategory = "input-category";
105
      const stringIdProductName = "product-name";
const stringIdDescription = "description";
106
107
108
       const stringIdShopName = "shop-name";
109
       const stringIdPrice = "price";
110
       const stringIdWebsite = "website";
       const stringIdImage = "image";
111
112
113
       // input elements:
114
       const inputContactName = document.getElementById(stringIdContactName);
115
       const inputEmail = document.getElementById(stringIdEmail);
       const inputTelephone = document.getElementById(stringIdTelephone);
116
117
       const inputCategory = document.getElementById(stringIdCategory);
118
       const inputProductName = document.getElementById(stringIdProductName);
       const inputDescription = document.getElementById(stringIdDescription);
119
120
       const inputShopName = document.getElementById(stringIdShopName);
       const inputPrice = document.getElementById(stringIdPrice);
121
       const inputWebsite = document.getElementById(stringIdWebsite);
122
       const inputImage = document.getElementById(stringIdImage);
123
124
125
       // help divs id strings
126
       const stringIdContactNameHelp = stringIdContactName + "-help";
      const stringIdEmailHelp = stringIdEmail + "-help";
const stringIdTelephoneHelp = stringIdTelephone + "-help";
127
128
129
       const stringIdCategoryHelp = stringIdCategory + "-help";
       const stringIdProductNameHelp = stringIdProductName + "-help";
130
       const stringIdDescriptionHelp = stringIdDescription + "-help";
131
132
       const stringIdShopNameHelp = stringIdShopName + "-help";
133
       const stringIdPriceHelp = stringIdPrice + "-help";
       const stringIdWebsiteHelp = stringIdWebsite + "-help";
134
       const stringIdImageHelp = stringIdImage + "-help";
135
136
137
138
       // help divs elements:
139
       const helpContactName = document.getElementById(stringIdContactNameHelp);
140
       const helpEmail = document.getElementById(stringIdEmailHelp);
141
       const helpTelephone = document.getElementById(stringIdTelephoneHelp);
142
       const helpCategory = document.getElementById(stringIdCategoryHelp);
```



version 2.13.4 There are 10 functior (https://github.com/jshint/j: Function with the larg

About median is 2.

Documentation (/docs) has 2.5.

nstall >

The most complex fu

Blog (/blog)

```
143
      const helpProductName = document.getElementById(stringIdProductNameHelp);
144
      const helpDescription = document.getElementById(stringIdDescriptionHelp);
                                                                                                  CONFIGURE
145
      const helpShopName = document.getElementById(stringIdShopNameHelp);
146
      const helpPrice = document.getElementById(stringIdPriceHelp);
147
      const helpWebsite = document.getElementById(stringIdWebsiteHelp);
148
      const helpImage = document.getElementById(stringIdImageHelp);
                                                                                                  version 2,13.4
There are 10 function
149
150
                                                                                           (https://github.com/jshint/j:
Function with the larg
151
      // regexes and validation error messages for failing the regex
152
      // regex email copied from javascripttutorial.net/javascript-dom/javascript-form-validatiout (median) is 2.
153
154
      // Example: name@domain.com
                                                                                            Documentation (/docs) has
155
                                                                                                  2.5.
      156
                                                                                                  The most complex fu
      var errorMessageEmail = "Please enter a valid email address for examle: name@domain.com";
157
                                                                                            Contribute (/contribute) edia
158
159
      // regex for UK phone number in any format copied from
      // https://stackoverflow.com/questions/11518035/regular-expression-for-gb-based-and-onlyling (/blog)
160
      // Example: 01279755875
161
      // Second Example: +441279755875
162
163
164
      var errorMessageTelephone = "Please enter a valid UK phone number (example: 01162983749)";
165
166
167
      // regex for Price copied from:
168
      // https://stackoverflow.com/questions/15958808/regexp-for-validating-price
169
      // Example: 1555,555.55
170
      const regexPrice = /^(d*([.,](?=\d{3}))?\d+)+((?!\2)[.,]\d\d)?$/;
171
172
      var errorMessagePrice = "Please enter price in american format for example 1,234.56 withou
173
174
      // regex for URL copied from:
175
      // https://stackoverflow.com/questions/161738/what-is-the-best-regular-expression-to-check
176
      // Example: http://www.google.com
177
178
      const regexURL = /((([A-Za-z]{3,9}:(?:\/\)?)(?:[-;:&=\+\$,\w]+@)?[A-Za-z0-9.-]+|(?:www.|[
      var errorMessageURL = "Please enter a valid URL for example: http://www.google.com";
179
180
181
      // debounce and instant feedback on input copied from the below link
182
      //https://www.javascripttutorial.net/javascript-dom/javascript-form-validation/
183
184
       st Function to delay response
185
186
187
      const debounce = (fn, delay = 500) => {
        let timeoutId;
188
189
        return (...args) => {
190
          // cancel the previous timer
191
          if (timeoutId) {
192
            clearTimeout(timeoutId);
193
194
          // setup a new timer
195
          timeoutId = setTimeout(() => {
            fn.apply(null, args);
196
          }, delay);
197
198
        };
199
      };
200
201
202
203
       * Gives instant feedback on input with the delay set above
204
       * some of the input fields are validated only for empty input,
       * others are validated using regex and checked if empty
205
206
      contactForm.addEventListener("input", debounce(function (e) {
207
208
        switch (e.target.id) {
209
          case stringIdContactName:
            isNotEmpty(inputContactName, helpContactName);
210
211
            break;
212
          case stringIdEmail:
            validateInput(inputEmail, helpEmail, regexEmail, errorMessageEmail);
213
214
            break:
```

```
215
           case stringIdTelephone:
             validateInput(inputTelephone, helpTelephone, regexTelephone, errorMessageTelephone);
216
                                                                                                               CONFIGURE
217
218
           case stringIdCategory:
             isNotEmpty(inputCategory, helpCategory);
219
220
             break;
                                                                                                       version 2,13.4
There are 10 function
(https://github.com/jshint/j:
Function with the larc
           case stringIdProductName:
221
222
             isNotEmpty(inputProductName, helpProductName);
223
             break;
           case stringIdDescription:
224
                                                                                                        About median is 2.
             isNotEmpty(inputDescription, helpDescription);
225
226
                                                                                                        Documentation (/docs) has
227
           case stringIdShopName:
                                                                                                               2.5. > The most complex fu
             isNotEmpty(inputShopName, helpShopName);
228
229
             break;
                                                                                                        Contribute (/contribute) edia
230
           case stringIdPrice:
231
             validateInput(inputPrice, helpPrice, regexPrice, errorMessagePrice);
                                                                                                        Blog (/blog)
232
233
           case stringIdWebsite:
234
              validateInput(inputWebsite, helpWebsite, regexURL, errorMessageURL);
235
             break;
236
           case stringIdImage:
             validateInput(inputImage, helpImage, regexURL, errorMessageURL);
237
238
239
         }
240
       }));
```