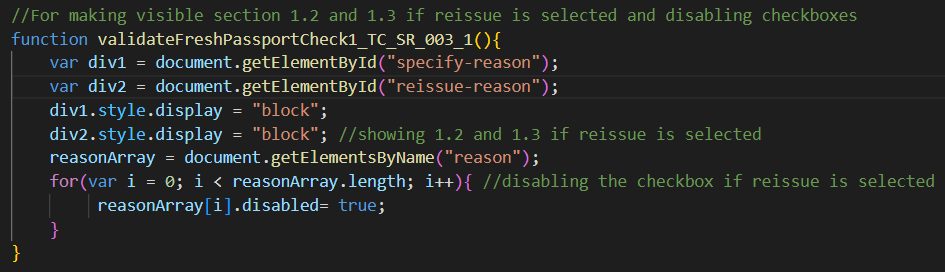


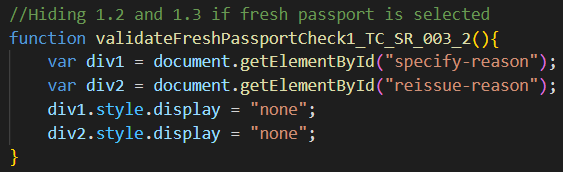
Algorithm: ***validateFreshPassportCheck1\_TC\_SR\_001()***

1. Start the function **validateFreshPassportCheck1\_TC\_SR\_001()**.
2. Use **document.querySelector** to get the selected radio button with the name "applying-for" and assign it to the variable **selected\_value1**.
3. If **selected\_value1** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to select an option for "applying for" and provide a hyperlink to the relevant section.
4. Else, if the value of selected\_value1 is "Re-issue-of-Passport":
   1. Use **document.querySelector** to get the selected radio button with the name "**re-issue reason**" and assign it to the variable **selected\_value2**.
   2. If **selected\_value2** is null or undefined (i.e., no radio button is selected):
5. Append a message to the variable message stating to select a reason as re-issue is selected and provide a hyperlink to the relevant section.
6. Else:
   1. Set the variable message to an empty string.
7. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
8. End the function.



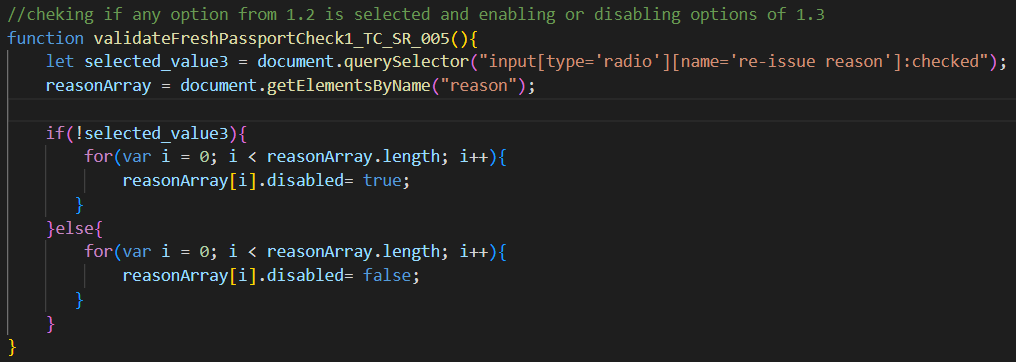
Algorithm: ***validateFreshPassportCheck1\_TC\_SR\_003\_1()***

1. Start the function **validateFreshPassportCheck1\_TC\_SR\_003\_1()**
2. Get the element with the ID "specify-reason" and assign it to the variable **div1**.
3. Get the element with the ID "reissue-reason" and assign it to the variable **div2**.
4. Set the display style of **div1** to "block".
5. Set the display style of **div2** to "block".
6. Get all elements with the name "reason" and assign them to the **reasonArray**.
7. Iterate over each element in **reasonArray** using a for loop:
8. Disable the current element by setting its disabled property to true.
9. End the function.



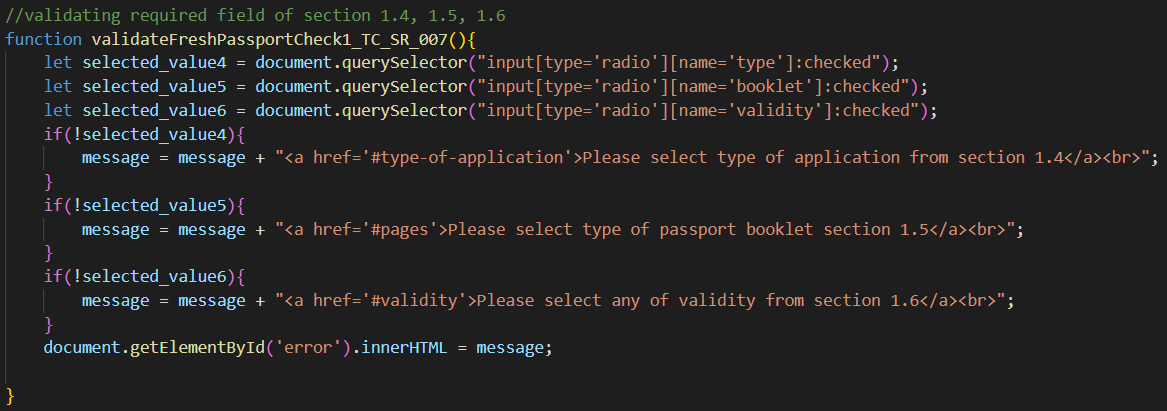
Algorithm: ***validateFreshPassportCheck1\_TC\_SR\_003\_2()***

1. Start the function **validateFreshPassportCheck1\_TC\_SR\_003\_2()**.
2. Get the element with the ID "specify-reason" and assign it to the variable **div1**.
3. Get the element with the ID "reissue-reason" and assign it to the variable **div2**.
4. Set the display style of **div1** to "none".
5. Set the display style of **div2** to "none".
6. End the function.

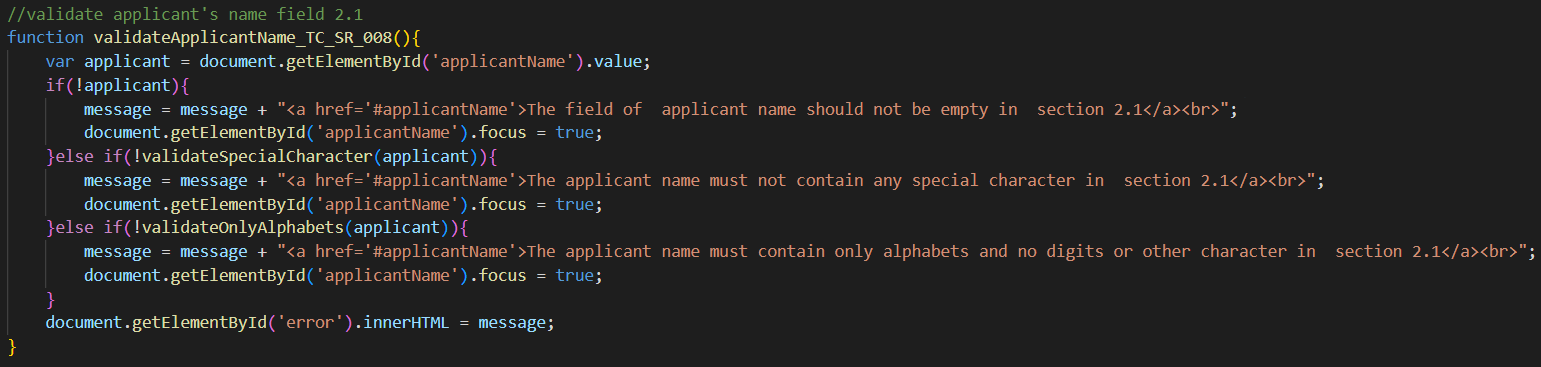


Algorithm: ***validateFreshPassportCheck1\_TC\_SR\_005()***

1. Start the function **validateFreshPassportCheck1\_TC\_SR\_005()**.
2. Use **document.querySelector** to get the selected radio button with the name "re-issue reason" and assign it to the variable **selected\_value3**.
3. Get all elements with the name "reason" and assign them to the **reasonArray**.
4. If **selected\_value3** is null or undefined (i.e., no radio button is selected):
   1. Iterate over each element in the **reasonArray** using a for loop:
      1. Disable the current element by setting its disabled property to true.
5. Else:
   1. Iterate over each element in the **reasonArray** using a for loop:
      1. Enable the current element by setting its disabled property to false.
6. End the function.

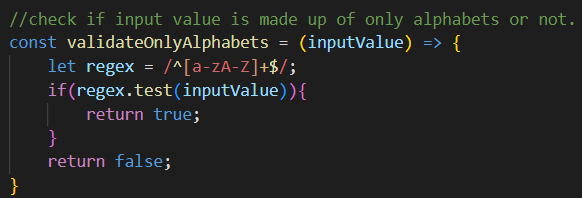
 Algorithm: ***validateFreshPassportCheck1\_TC\_SR\_007()***

1. Start the function **validateFreshPassportCheck1\_TC\_SR\_007()**.
2. Use **document.querySelector** to get the selected radio button with the name "type" and assign it to the variable **selected\_value4**.
3. Use **document.querySelector** to get the selected radio button with the name "booklet" and assign it to the variable **selected\_value5**.
4. Use **document.querySelector** to get the selected radio button with the name "validity" and assign it to the variable **selected\_value6**.
5. If **selected\_value4** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select the type of application from **section 1.4”** and provide a hyperlink to the relevant section.
6. If **selected\_value5** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select the type of passport booklet from **section 1.5”** and provide a hyperlink to the relevant section.
7. If **selected\_value6** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select any validity option from **section 1.6**” and provide a hyperlink to the relevant section.
8. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
9. End the function.



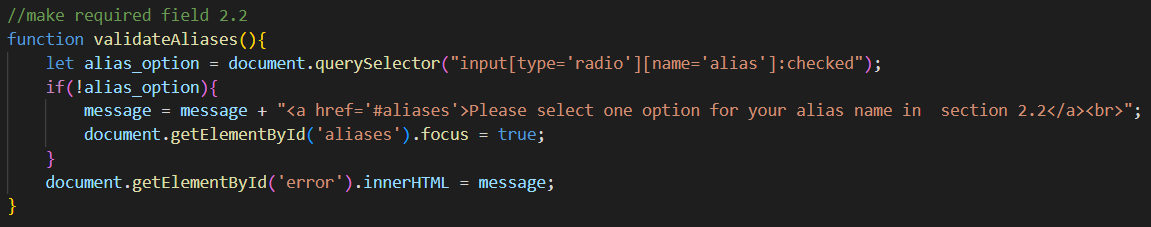
Algorithm: ***validateApplicantName\_TC\_SR\_008()***

1. Start the function **validateApplicantName\_TC\_SR\_008().**
2. Get the value of the element with the ID 'applicantName' and assign it to the variable **applicant**.
3. If the **applicant** value is empty or null:
   1. Append a message to the variable message stating that the “applicant name field should not be empty in section 2.1” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'applicantName' to true.
4. Else, if the applicant value contains any special character:
   1. Append a message to the variable message stating that “the applicant name must not contain any special characters in section 2.” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'applicantName' to true.
5. Else, if the applicant value contains digits or other non-alphabetic characters:
   1. Append a message to the variable message stating that “the applicant name must contain only alphabets and no digits or other characters in section 2.1” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'applicantName' to true.
6. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
7. End the function.



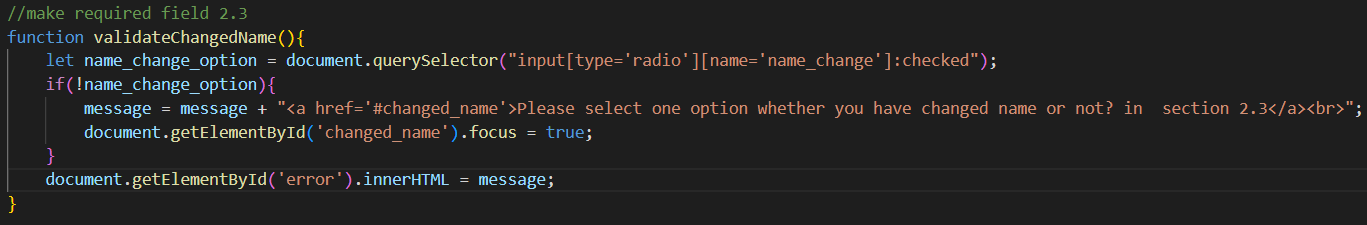
Algorithm: ***validateOnlyAlphabets(inputValue)***

1. Start the function **validateOnlyAlphabets** with **inputValue** as the parameter.
2. Create a regular expression pattern “**/^[a-zA-Z]+$/”** and assign it to the variable **regex**.
3. If **regex.test(inputValue)** returns true (i.e., the **inputValue** consists of only alphabets):
   1. Return true.
4. Return false.
5. End the function.



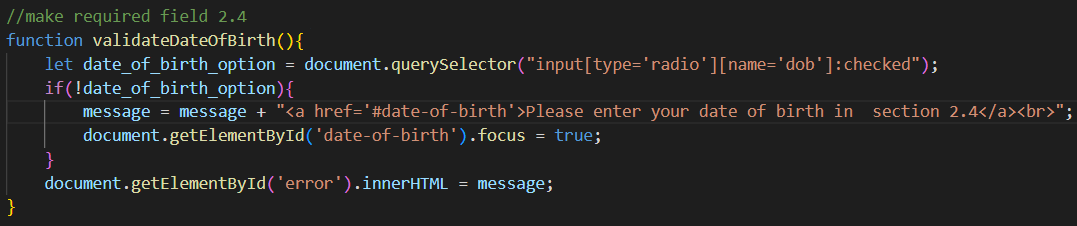
Algorithm: ***validateAliases()***

1. Start the function **validateAliases().**
2. Use **document.querySelector** to get the selected radio button with the name "alias" and assign it to the variable **alias\_option**.
3. If **alias\_option** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select one option for your alias name in section 2.2” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'aliases' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



Algorithm*:* ***validateChangedName()***

1. Start the function **validateChangedName().**
2. Use **document.querySelector** to get the selected radio button with the name "**name\_change**" and assign it to the variable **name\_change\_option**.
3. If **name\_change\_option** is null or undefined (i.e., no radio button is selected):
   1. Append the error message stating that “select one option whether you have changed name or not? in section 2.3” to the variable message.
   2. Set the focus of the element with the ID 'changed\_name' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



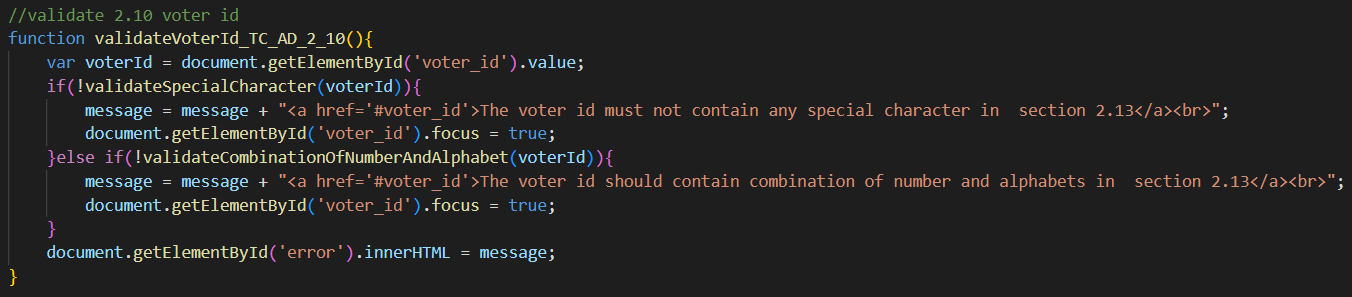
Algorithm: ***validateDateOfBirth()***

1. Start the function **validateDateOfBirth()**.
2. Retrieve the selected radio button element with the name attribute 'dob' using the **document.querySelector** method and assign it to the variable **date\_of\_birth\_option**.
3. If **date\_of\_birth\_option** is null or undefined (i.e., no radio button is selected):
   1. Append the error message "<a href='#date-of-birth'>Please enter your date of birth in section 2.4</a><br>" to the variable message.
   2. Set the focus of the element with the ID 'date-of-birth' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



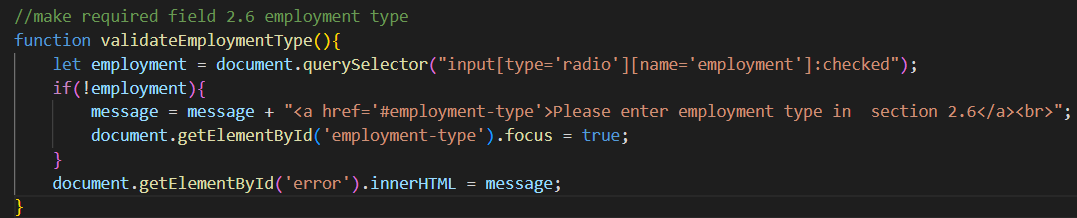
Algorithm: ***validatePlaceOfBirth\_TC\_AD\_2\_5()***

1. Start the function **validatePlaceOfBirth\_TC\_AD\_2\_5().**
2. Retrieve the values of the input fields with IDs 'city', 'district', 'state', and 'country' and assign them to the respective variables **city, district, state, and country**.
3. Check if the **city** field is empty:
   1. Append the error message "<a href='#city'>Please enter your city of place of birth in section 2.5</a><br>" to the variable message.
   2. Set the focus of the **'city'** element to true.
4. If the **city** field is not empty, check if it contains any special characters:
   1. Append the error message "<a href='#city'>Please do not enter any special character in the city of place of birth in section 2.5</a><br>" to the variable message.
   2. Set the focus of the **'city'** element to true.
5. If the **city** field does not contain special characters, check if it contains only alphabets:
   1. Append the error message "<a href='#city'>Please enter only alphabets in the **city** of place of birth in section 2.5</a><br>" to the variable message.
   2. Set the focus of the **'city'** element to true.
6. Repeat steps 3-5 for the fields **'district'**, **'state'**, and **'country'**.
7. Check if the **'country'** field is not equal to "**INDIA**" or "**Undivided India**":
   1. Append the error message "<a href='#country'>Please enter valid country and details of place of birth in section 2.5</a><br>" to the variable message.
   2. Set the focus of the **'country'** element to true.
8. Check if all the place of birth fields (**city, district, state, and country**) have the same value:
   1. Append the error message "<a href='#city'>Please enter valid place of birth in section 2.5</a><br>" to the variable message.
   2. Set the focus of the **'city'** element to true.
9. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
10. End the function.



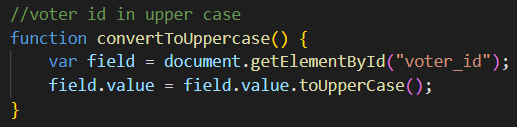
Algorithm: ***validateVoterId\_TC\_AD\_2\_10()***

1. Start the function **validateVoterId\_TC\_AD\_2\_10().**
2. Retrieve the value of the input field with the ID 'voter\_id' and assign it to the variable **voterId**.
3. Check if the **voterId** contains any special characters:
   1. Append the error message "<a href='#voter\_id'>The voter id must not contain any special character in section 2.13</a><br>" to the variable message.
   2. Set the focus of the **'voter\_id'** element to true.
4. If the **voterId** does not contain special characters, check if it contains a combination of numbers and alphabets:
   1. Append the error message "<a href='#voter\_id'>The voter id should contain a combination of numbers and alphabets in section 2.13</a><br>" to the variable message.
   2. Set the focus of the **'voter\_id'** element to true.
5. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
6. End the function.



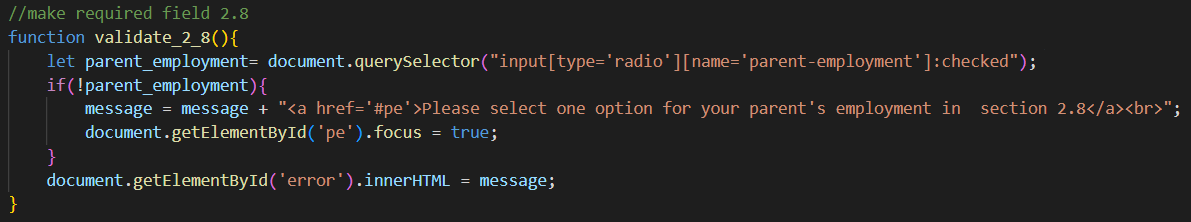
Algorithm: ***validateEmploymentType()***

1. Start the function **validateEmploymentType()**.
2. Use **document.querySelector** to get the selected radio button with the name "employment" and assign it to the variable **employment**.
3. If **employment** is null or undefined (i.e., no radio button is selected):
   1. Append the error message "<a href='#employment-type'>Please enter employment type in section 2.6</a><br>" to the variable message.
   2. Set the focus of the element with the ID 'employment-type' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



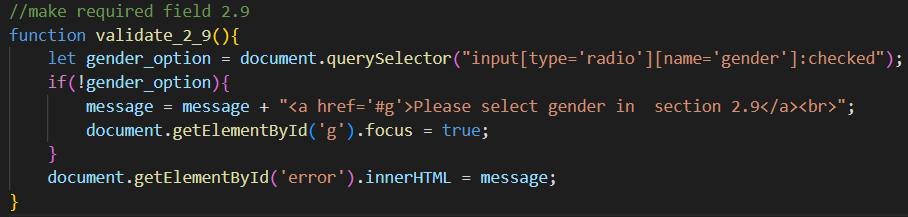
Algorithm: ***convertToUppercase()***

1. Start the function **convertToUppercase().**
2. Get the element with the ID "voter\_id" and assign it to the variable **field**.
3. Get the current value of the **field** using **field.value**.
4. Convert the value to uppercase using the **toUpperCase()** method.
5. Set the modified value back to the **field** using **field.value = modifiedValue**.
6. End the function.



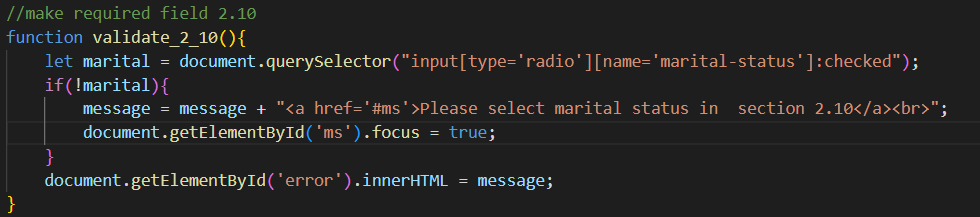
Algorithm***: validate\_2\_8()***

1. Start the function **validate\_2\_8()**.
2. Use **document.querySelector** to get the selected radio button with the name "parent-employment" and assign it to the variable **parent\_employment**.
3. If **parent\_employment** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select one option for your parent's employment in section 2.8” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'pe' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



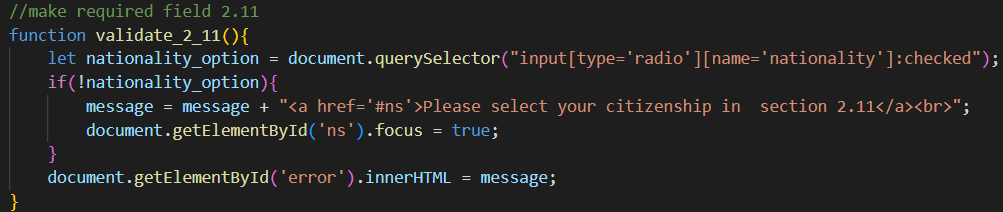
Algorithm: ***validate\_2\_9()***

1. Start the function **validate\_2\_9()**.
2. Use **document.querySelector** to get the selected radio button with the name "gender" and assign it to the variable **gender\_option**.
3. If **gender\_option** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select gender in section 2.9” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'g' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



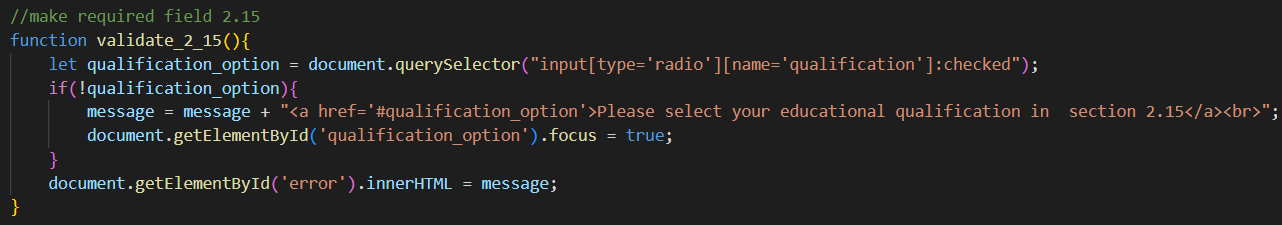
Algorithm: ***validate\_2\_10()***

1. Start the function **validate\_2\_10().**
2. Use **document.querySelector** to get the selected radio button with the name "marital-status" and assign it to the variable **marital**.
3. If **marital** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select marital status in section 2.10” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'ms' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



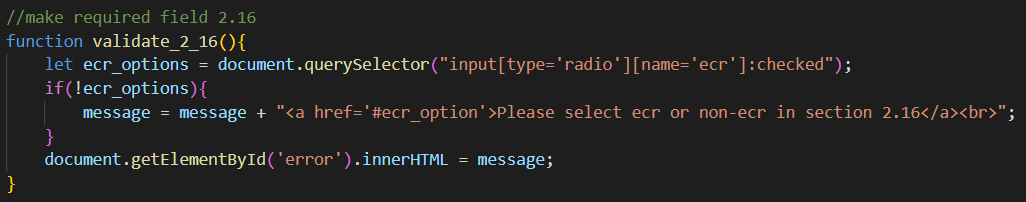
Algorithm: **validate\_2\_11()**

1. Start the function **validate\_2\_11().**
2. Use **document.querySelector** to get the selected radio button with the name "nationality" and assign it to the variable **nationality\_option**.
3. If **nationality\_option** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select citizenship in section 2.11” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'ns' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



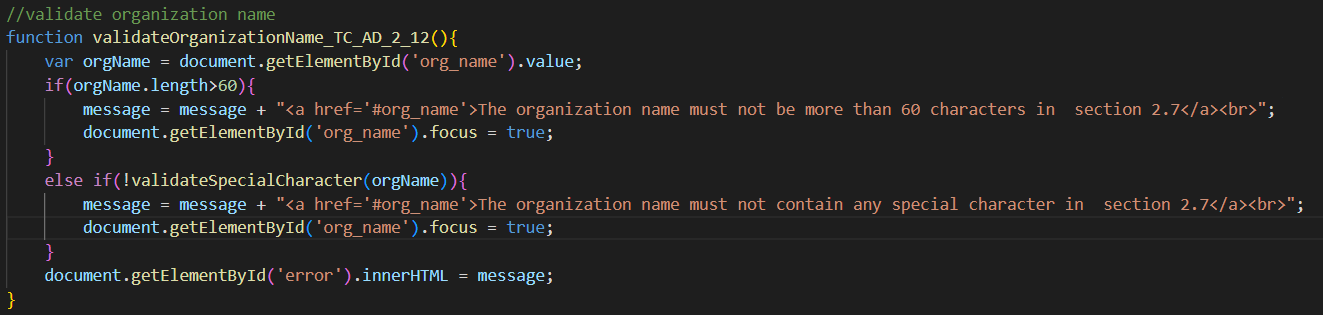
Algorithm: ***validate\_2\_15()***

1. Start the function **validate\_2\_15().**
2. Use **document.querySelector** to get the selected radio button with the name "qualification" and assign it to the variable **qualification\_option**.
3. If **qualification\_option** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select educational qualification in section 2.15” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'qualification\_option' to true.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



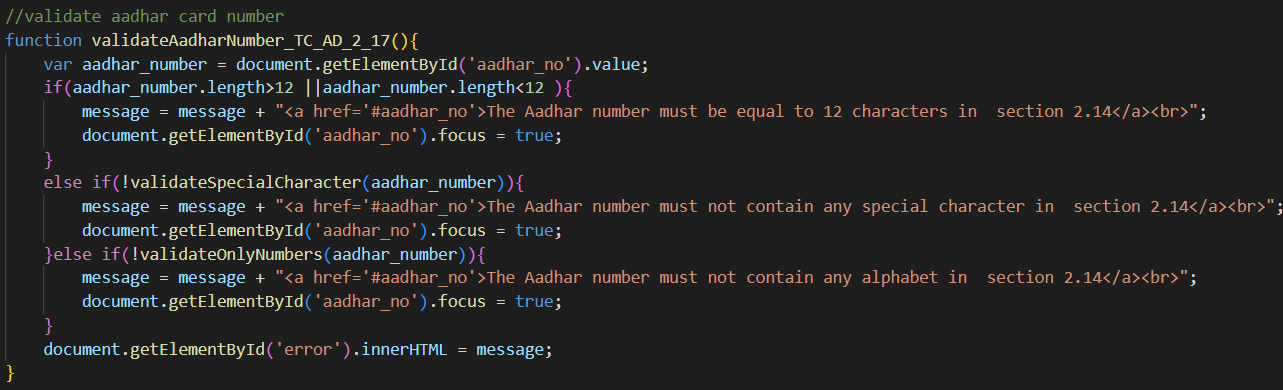
Algorithm: ***validate\_2\_16()***

1. Start the function **validate\_2\_16().**
2. Use **document.querySelector** to get the selected radio button with the name "ecr" and assign it to the variable **ecr\_options**.
3. If **ecr\_options** is null or undefined (i.e., no radio button is selected):
   1. Append a message to the variable message stating to “select ECR or non-ECR in section 2.16” and provide a hyperlink to the relevant section.
4. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
5. End the function.



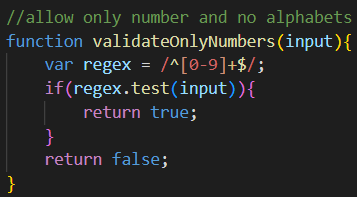
Algorithm: ***validateOrganizationName\_TC\_AD\_2\_12()***

1. Start the function **validateOrganizationName\_TC\_AD\_2\_12()**.
2. Get the value of the element with the ID 'org\_name' and assign it to the variable **orgName**.
3. If the length of **orgName** is greater than 60:
   1. Append a message to the variable message stating that “the organization name must not exceed 60 characters in section 2.7” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'org\_name' to true.
4. Else if **orgName** contains any special characters:
   1. Append a message to the variable message stating that “the organization name must not contain any special characters in section 2.7” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'org\_name' to true.
5. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
6. End the function.



Algorithm: ***validateAadharNumber\_TC\_AD\_2\_17()***

1. Start the function **validateAadharNumber\_TC\_AD\_2\_17().**
2. Get the value of the element with the ID 'aadhar\_no' and assign it to the variable **aadhar\_number**.
3. If the length of **aadhar\_number** is greater than 12 or less than 12:
   1. Append a message to the variable message stating that “the Aadhar number must be equal to 12 characters in section 2.14” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'aadhar\_no' to true.
4. Else if **aadhar\_number** contains any special characters:
   1. Append a message to the variable message stating that “the Aadhar number must not contain any special characters in section 2.14” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'aadhar\_no' to true.
5. Else if **aadhar\_number** contains any alphabet characters:
   1. Append a message to the variable message stating that “the Aadhar number must not contain any alphabet characters in section 2.14” and provide a hyperlink to the relevant section.
   2. Set the focus of the element with the ID 'aadhar\_no' to true.
6. Set the innerHTML of the element with the ID 'error' to the value of the variable message.
7. End the function.



Algorithm: ***validateOnlyNumbers(input)***

1. Start the function **validateOnlyNumbers(input)** and accept an input parameter.
2. Initialize the variable **regex** with the regular expression “**/^[0-9]+$/**”, which matches only numeric characters.
3. Check if the input matches the **regex** using the **test()** method:
   1. If the input matches the **regex**, i.e., consists only of numeric characters:
      1. Return true to indicate that the input contains only numbers.
   2. Otherwise:
      1. Return false to indicate that the input does not contain only numbers.
4. End the function.