1.JS - Withdraw and Deposit

Content :

H1 tag -Bank

H2 tag -Amount withdraw and deposit

Button tag - Withdraw

Deposit

Constraints :

Design the html page as given in the sample screenshot1.

Tag 'h1' must be present for minimum of 1 time in the html page.

Tag 'h2' must be present for minimum of 1 time in the html page.

Tag 'input' must be present for minimum of 1 time in the html page.

Tag 'button' must be present for minimum of 2 times in the html page.

Tag 'p' must be present for minimum of 1 time in the html page.

The 'input' tag id should be 'number'.

First 'button' tag id should be 'button1' and the onClick function should be named 'Withdraw( )'

Second 'button' tag id should be 'button2' and the onClick function should be named 'Deposit( )'

Conditions :

Initially the amount must be 1000 in the 'p' tag with id 'amount'.

If user clickswithdraw button after providing input in the id 'number'. Subtract the amount from the total amount.

If user clicks deposit button after providing input in the id 'number'. Add the amount to the total amount.Calculate the amount based on action performed by the user and display total amount in the 'p' tag with id 'amount'.

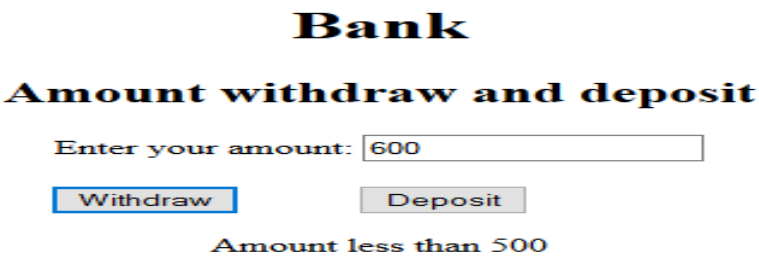
Minimum amount must be 500.If the balance amount is less-than 500 display "Amount less than 500" in the 'p' tag with id 'amount'.

Sample Screenshot 1 :

Sample Screenshot 2:

Sample Screenshot 3 :

Sample Screenshot 5:



2.JS - Calculator

Hints :

Function

A JavaScript function is a block of code designed to perform a particular task.A JavaScript functi on is executed when "something"
invokes it (calls it/onclick).

Example :

```
function myFunction(p1, p2) {
 return p1 * p2;
}
```

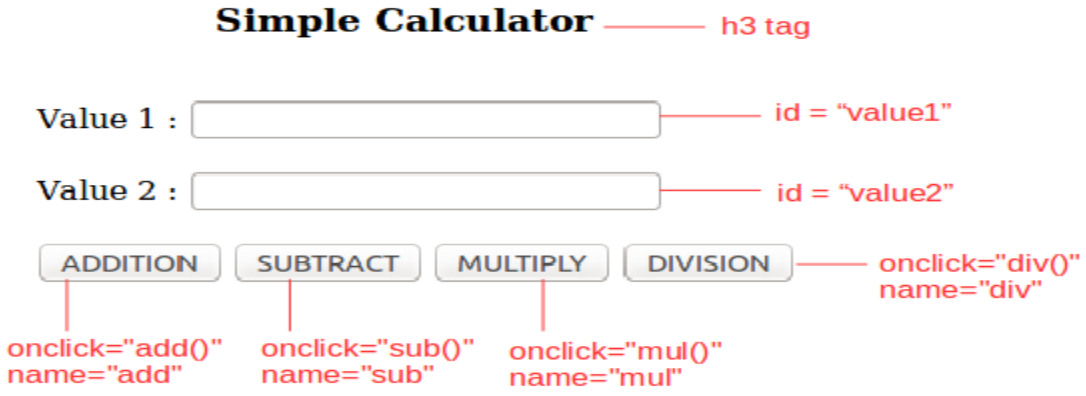
Constraints :

File name should be index.html.
The first input field should have id "value1"
The second input field should have id "value2"
The div element must be with id "result" to display the message.
The functions named add(), sub(), mul() and div() must be present to perform the operations like addition, subtraction, multiplication and division for the buttons respectively.

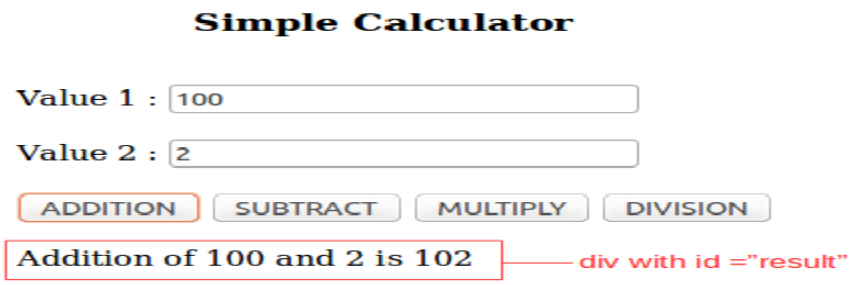
Note :

Content of the page should be present as shown in the screenshot.

Sample Screenshot 1:



Sample Screenshot 2:



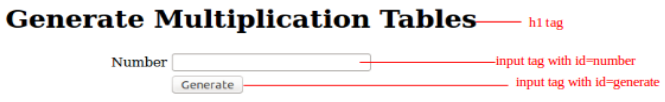
3.**Multiplication Table**

Hints:
For loop
For loop enable executing one or more statements repeatedly for several number of times.
It loops through a block of code a number of times.
Syntax:

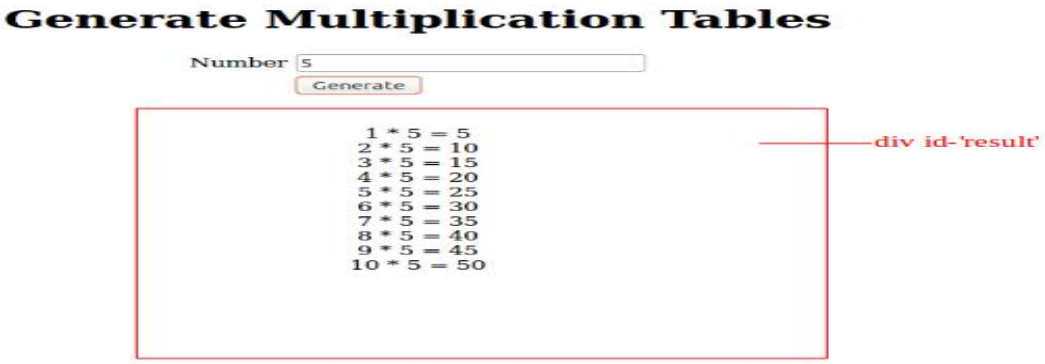
```
for (statement 1; statement 2; statement 3)
{
  code block to be executed
}
```

Have one text box to get the number from the user and generate the multiplication tables for 10 iterations.
Design the web page as shown in the sample screenshot 1.
Constraints :
File name should be index.html.
Have an input field with id 'number'.
Have another input field of type button with id 'generate'.
Have a div with id "result" to display the message.
Create a function named generateTable() to display the welcome message.

Sample Screenshot 1:



Sample Screenshot 2:



4.JS Calculation Using Boolean Condition

Hints :
A JavaScript Boolean represents one of two values: **true** or **false**.
Eg :

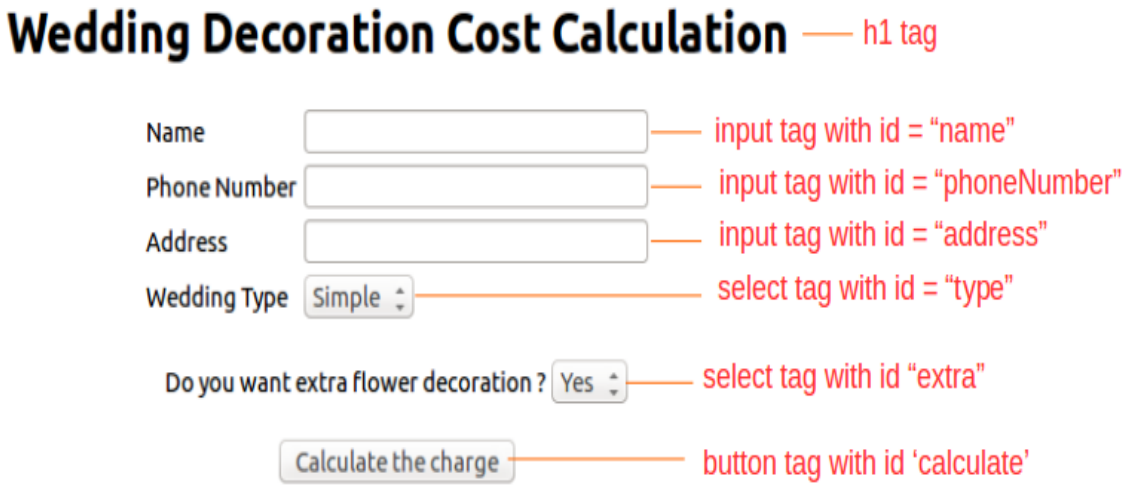
```
var x = true;
if (Boolean(x)) {
// true condition
}
else {
// false condition
}
```

Constraints :
File name should be 'index.html'
Inline script must be used.
Refer the screenshots for id specifications.
For the select box with id 'type', have "Simple" and "Grand" as two options.
For the select box with id 'extra', have "Yes" and "No" as two options.
Create a String variable.
If extra flower decoration is needed (i.e., 'Yes'), assign the value of the variable as 'true' and proceed it with using Boolean condition.
If extra flower decoration is not needed (i.e., 'No'), assign the value of the variable as 'false' and proceed it with using Boolean condition.

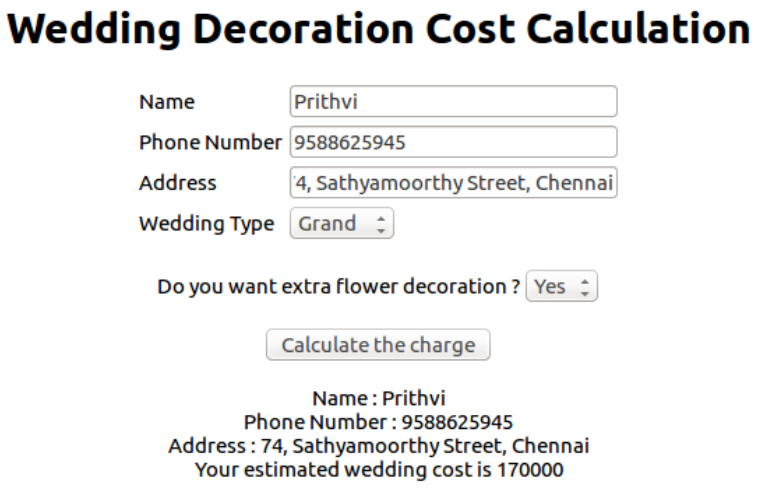
Conditions :
For the 'Grand' type, the cost is '150000'.
For the 'Simple' type, the cost is '50000'.

If the extra flower decoration is needed (i.e., 'Yes') add '20000' with the cost.

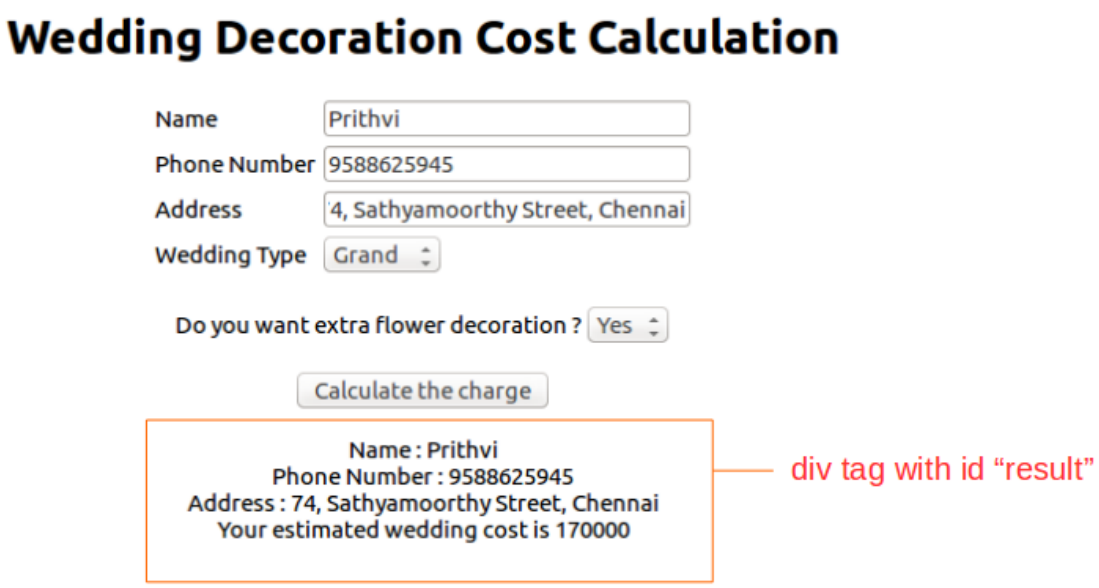
Sample Screenshot 1



Sample Screenshot 3 :



Sample Screenshot 4 :



5.Byte Converter

Hints :

charCodeAt()

The charCodeAt() method returns the Unicode of the character at the specified index in a string. The index of the first character is 0, the second character 1, and so on.

Syntax :

string.charCodeAt(index)

Eg :

The input String is **hello**

h - 104

e - 101

l - 108

l - 108

0 - 111

The converted value is **104101108108111**

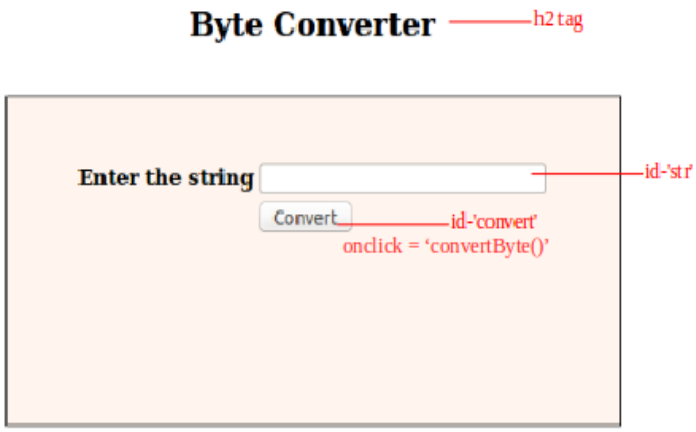
Constraints:

- Design the web page as shown in the sample screenshot 1.
- Kindly follow the naming convension as mentioned in the screenshot.
- Convert the given string to byte array and display it in the div with id 'result'.
-

Include the below function / method in the script.

S.No	Method Name	Description
1	convertByte()	This method is used to convert the string into the byte array.

Sample Screenshot 1 :



Sample Screenshot 2 :

6.JS Palindrome

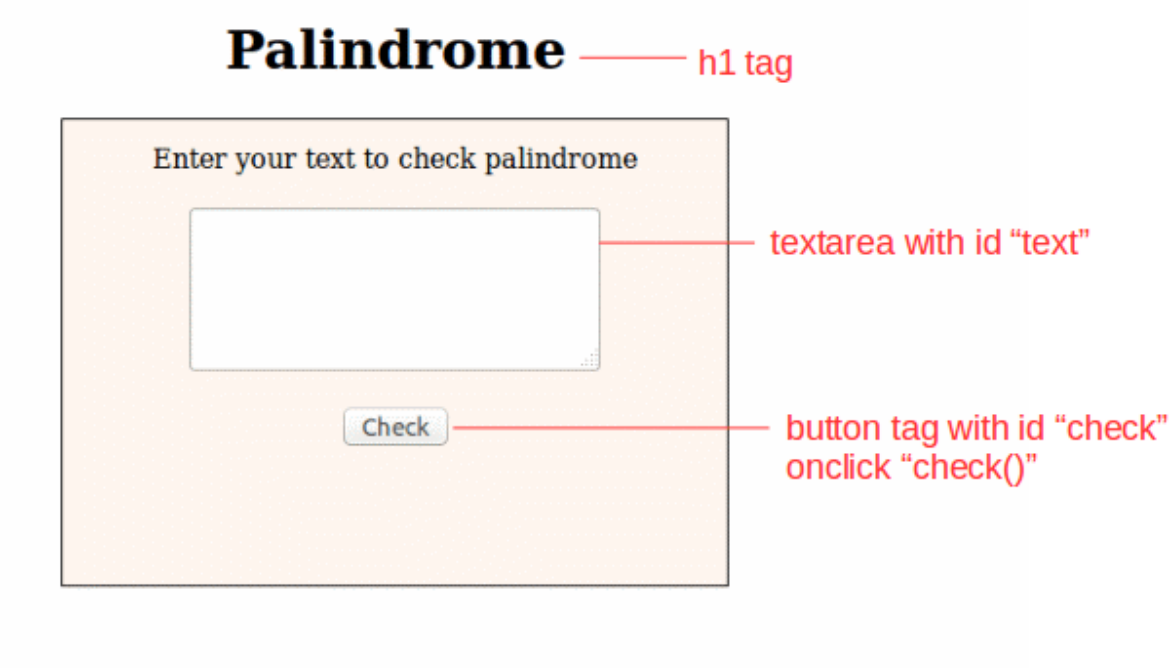
Constraints :
File name should be index.html.
Design the html page as given in the sample screenshot1.
Tag 'h1' must be present in the html page.
The textarea with id 'text' must be present in the html page.
Button tag must be present in the html page.
The 'onclick' functions for the Check button is 'check()' and id is 'check'.
The 'div' tag must be present with the id 'result'.

Conditions :
If the string has space or non alphanumeric characters, remove both space and a non alphanumeric characters and validate.
If the string has UpperCase, then convert it into LowerCase and validate.
If the string is palindrome, then print 'text is a palindrome' else print 'text is not a palindrome'.
Include the function/method in the script

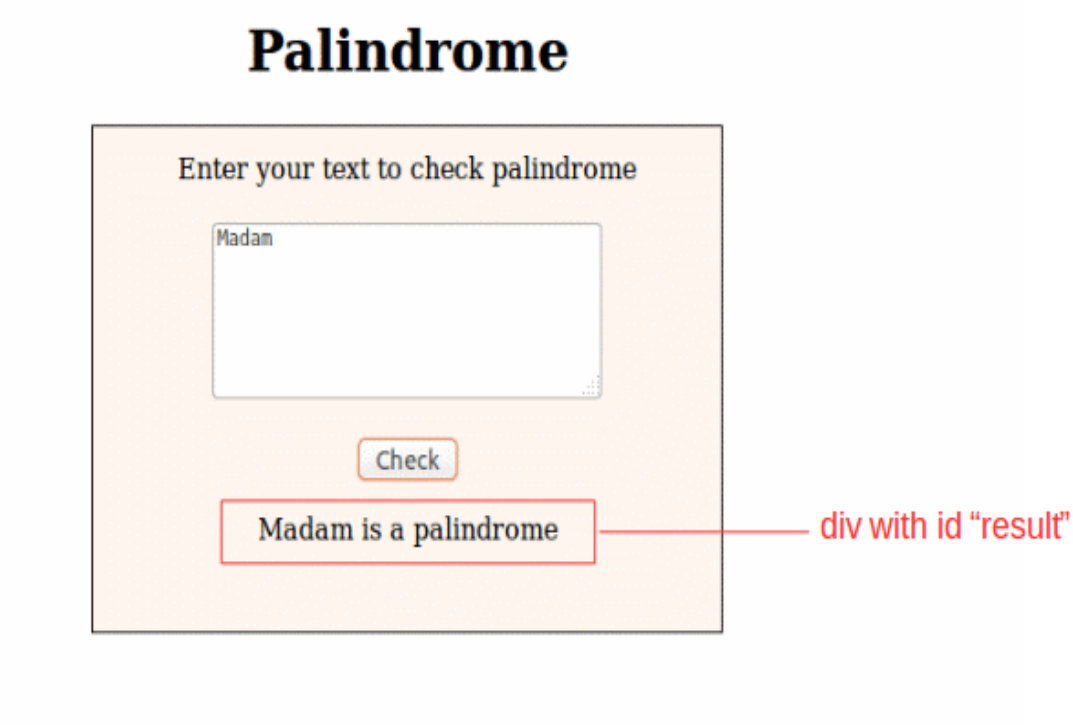
S.No	Method Name	Description
1	check()	This method is used to check the input string is palindrome or not.

Note :
Content of the page should be present as shown in the screenshot.
Kindly refer the content which is given as a part of description.

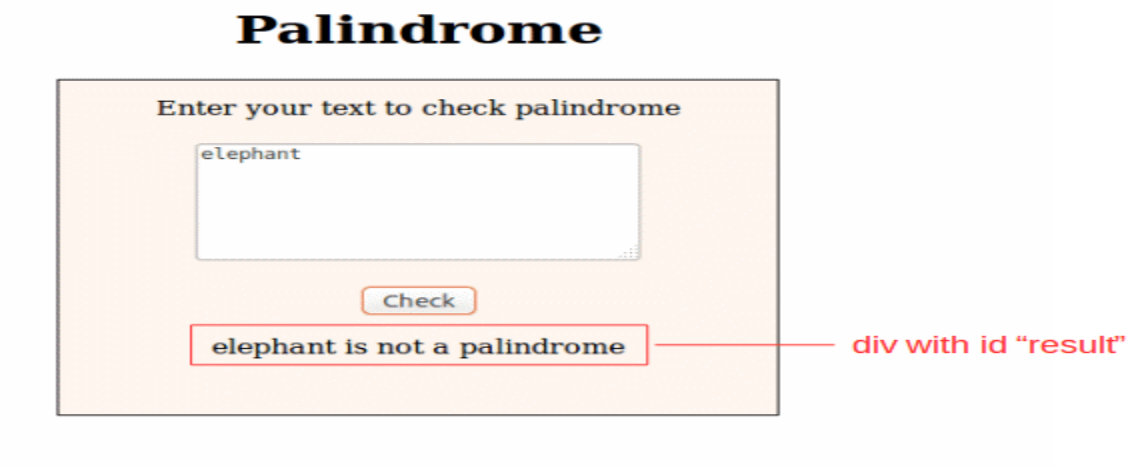
Sample Screenshot 1 :



Sample Screenshot 2 :



Sample Screenshot 3 :



7.Match and equals function

Constraints :

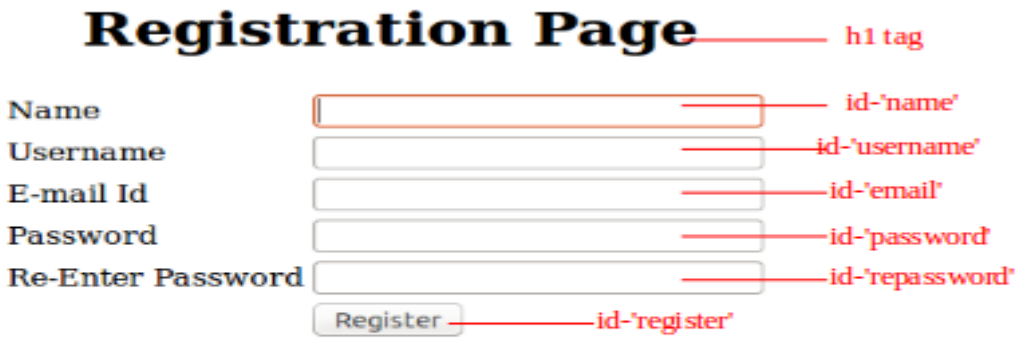
- Design the web page as shown in the sample screenshot 1.
- Kindly follow the naming convention as mentioned in the screenshot
- The fields for email id, password and re-enter password should not be empty.
- Check if the Password and Re-Enter Password fields are same
- Validate the e-mail id entered by the user.
- Kindly refer the screenshots for custom error messages.

Include the below function / method in the script.

S.No	MethodName	Description
1.	registration()	This method is used to validate the following fields email , password and Re-Enter Password. If the validation is successful print the sucess message 'Login Successful' in h3 tag.

Note :
Content of the page should be present as shown in the screenshot.
Kindly refer the content which is given as a part of description.

Sample Screenshot 1 :



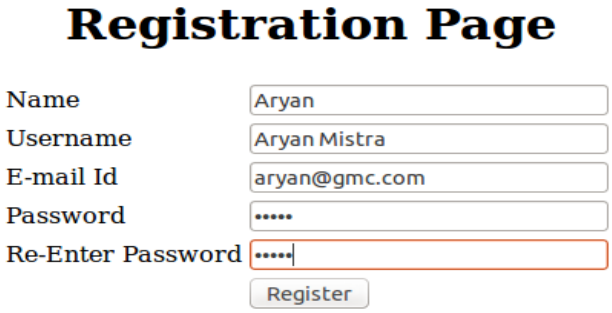
Sample Screenshot 2 :

If the input fields are empty,

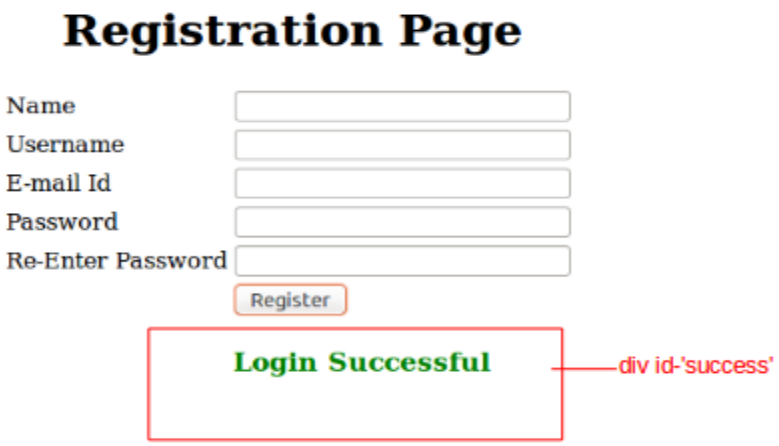
Sample Screenshot 3 :

If they entered invalid email id or mismatching passwords

Sample Screenshot 4 :



Sample Screenshot 5 :



8.Email Validation

Constraints :
Design the web page as shown in the sample screenshot 1.
Create an input field with id 'email'.
Create a div element with id 'result'.
Create a button with id 'validate' and onclick="validate()"
If the E-mail is valid display "Provided E-mail Id is Valid" inside the div element with id "result" in green color
If the E-mail is invalid display "Invalid E-mail. Please provide a valid E-mail id" inside the div element with id "result" in red color.

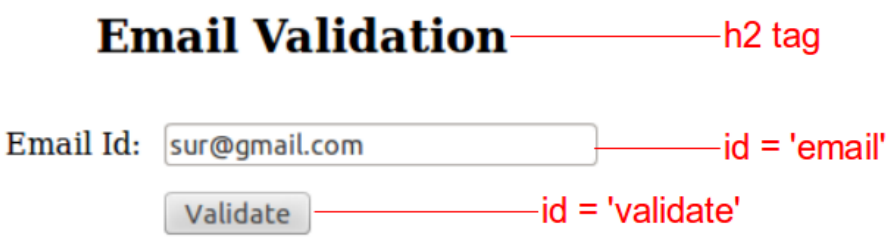
Conditions :
The email id must contains @ and . (dot)

Note :
Content of the page should be present as shown in the screenshot.

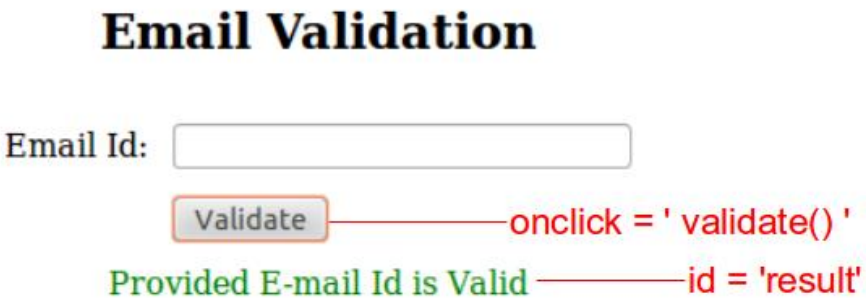
Kindly refer the content which is given as a part of description.

Sample Screenshot 1 :

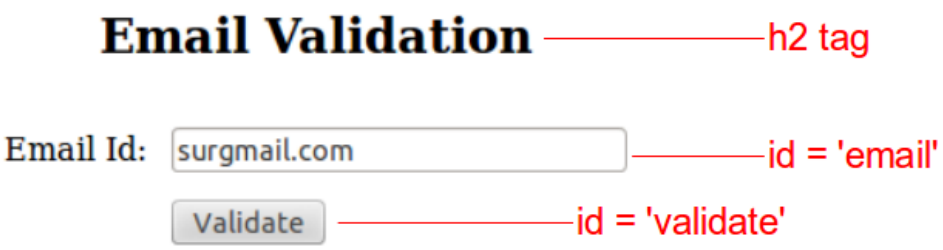
Sample Screenshot 2 :



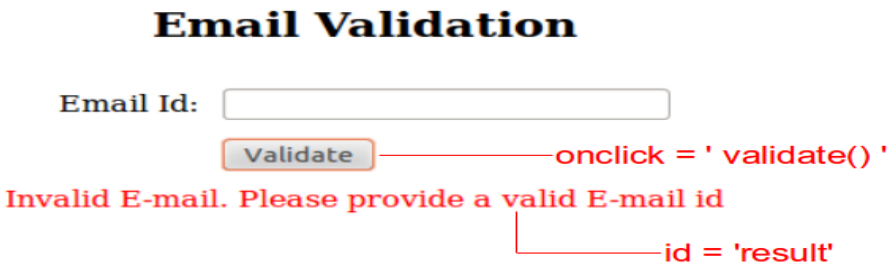
Sample Screenshot 3 :



Sample Screenshot 4 :



Sample Screenshot 5 :



8.Password Validation

Hints :

test() method

The test() method tests for a match in a string. This method returns true if it finds a match, other wise it returns false.

Syntax :

element.test(string)

Pattern Matching :

The pattern attribute specifies a regular expression that the <input> element's value is checked ag ainst.

[abc] - Find any of the characters between the brackets

[0-9] - Find any of the digits between the brackets

[!@#$%^&*] - Find any of the special characters between the brackets

^ - Start of the expression
$ - End of the expression

Example:

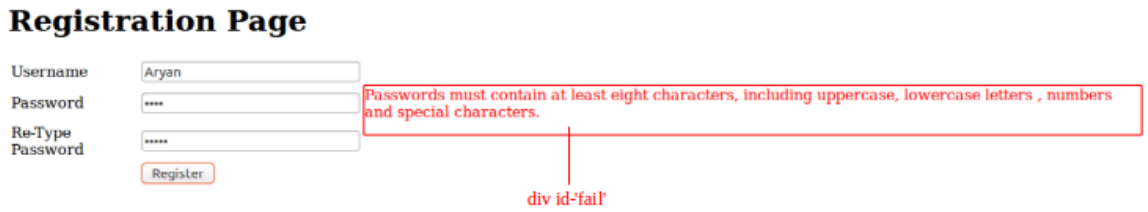
```
var string = "sample1";
var re = new RegExp("^([a-z0-9]{5,})$");
if (re.test(string)) {
   console.log("Valid");
} else {
   console.log("Invalid");
}
```

Constraints :

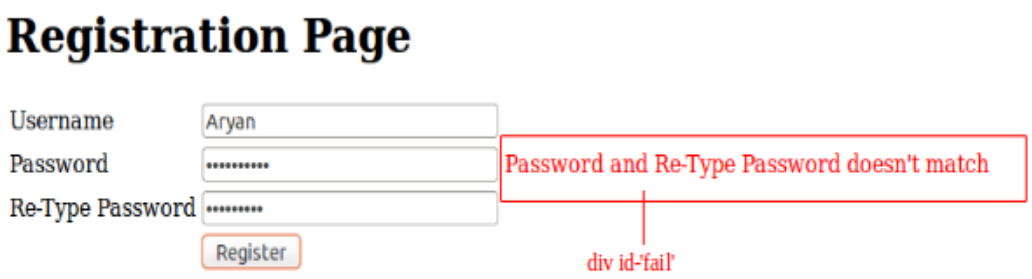
- Design the web page as shown in the sample screenshot 1.
- Create an input field with id 'name' and type 'text'.
- Create an input field with id 'password' and type 'password'.
- Create an input field with id 'repassword' and type 'password'.
- Display the success message in the div element with id 'success' and provided styling with color green.
- Display the invalid message in the div element with id 'fail' and provided styling with color red.
- Create a button with id 'register', type 'button', and onclick "registration()".
- If the password do not match the given condition display the error message 'Passwords must contain at least eight characters,including uppercase,lowercase letters,numbers and special characters.'
- If the password and re-enter password does not match display the following error message 'Password and Re-Type Password doesn't match'
- If the password and re-enter password is correct then display 'Login Successful'.

Sample Screenshot 1 :

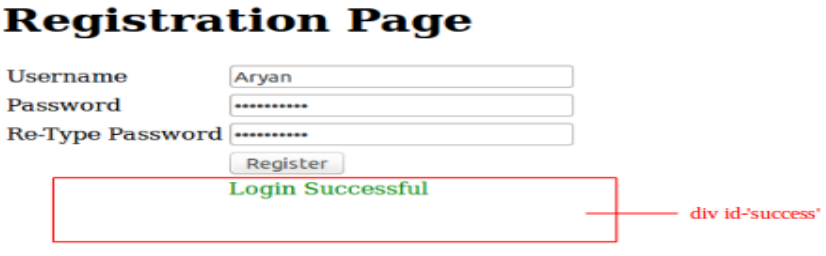
Sample Screenshot 2 :



Sample Screenshot 3 :



Sample Screenshot 4 :



Form validation

Hints : test() method The test() method tests for a match in a string. This method returns true if it finds a match, otherwise it returns false. Syntax : element.test(string) Pattern Matching : The pattern attribute specifies a regular expression that the <input> element's value is checked against. [abc] - Find any of the characters between the brackets [0-9] - Find any of the digits between the brackets [!@#$%^&*] - Find any of the special characters between the brackets ^ - Start of the expression $ - End of the expression Example: var string = "sample1"; var re = new RegExp("^**([a-z0-9]{5,})$**"); if (re.test(string)) { console.log("Valid"); } else { console.log("Invalid"); }

Constraints :

Design the web page as shown in the sample screenshot 1.
The h2 tag nust be present as given in the screenshot.
Create the input fields with the ids given in the screenshot.
Give the onchange="fieldValidate()" in all the input fields for validations.
Create a button with id 'register' and onclick="details()"
Print the respective error messages for all the feilds with the respective ids as given in the screenshot.
When the details are entered correctly, then display the details of the customer along with the success message in the respective ids as given in the screenshot 5.

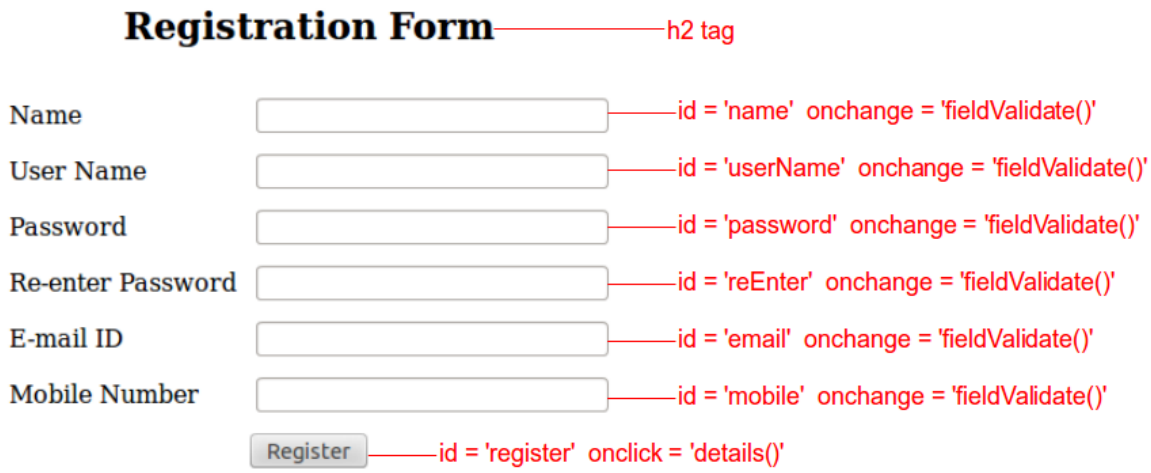
Conditions :

- All the fields should not be left blank.
- Passwords must contain atleast eight characters, including uppercase, lowercase, numbers and special characters.
- Password and Re-Type Password should be equal.
- E-mail should be in the format**(string@string.string(2 to 4 characters))** Eg: maya@gmail.com
- Mobile number should have 10 digits.

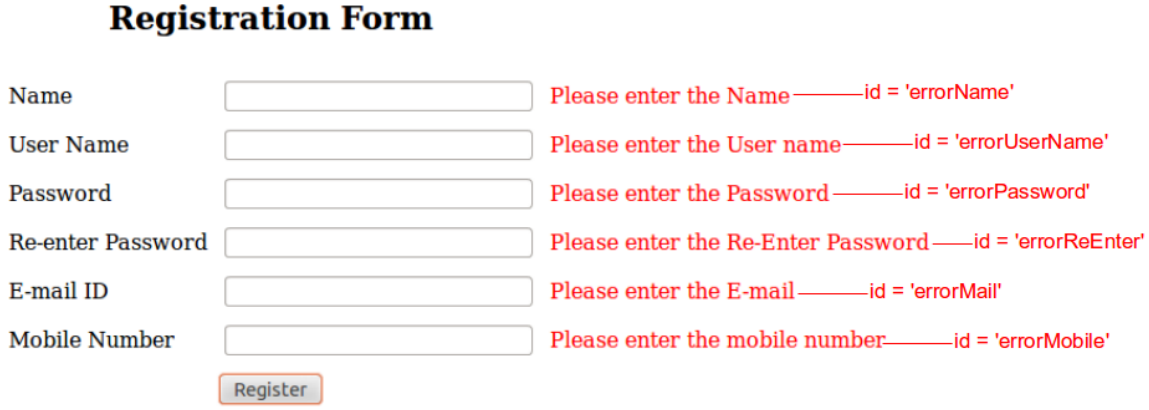
Note :

Content of the page should be present as shown in the screenshot.
Kindly refer the content which is given as a part of description.

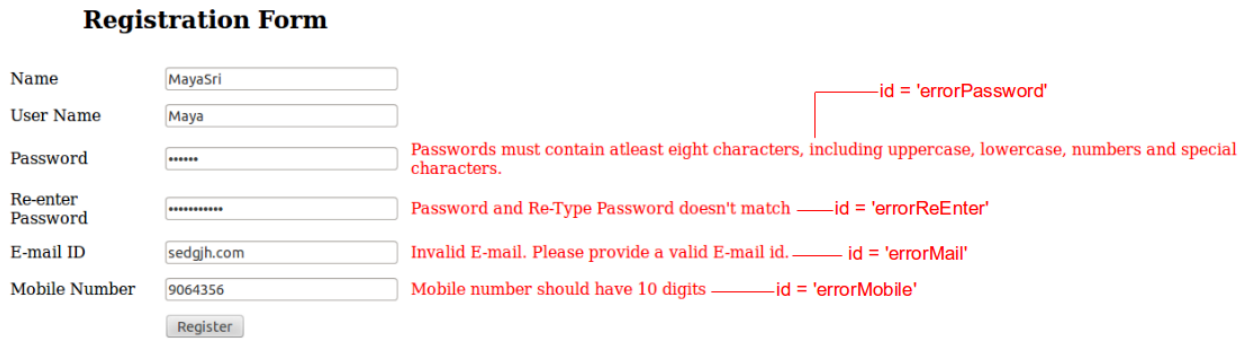
Sample Screenshot 1 :



Sample Screenshot 2 :

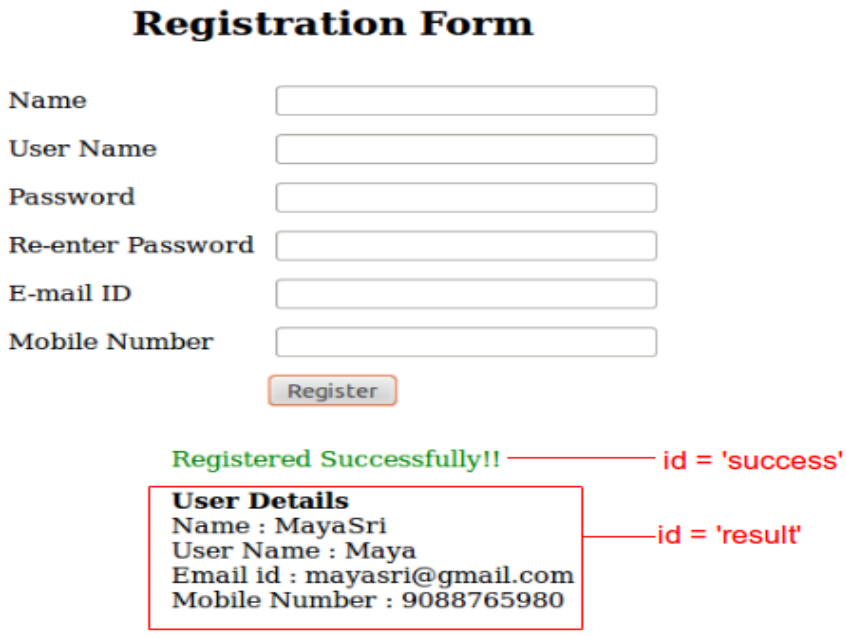


Sample Screenshot 3 :



Sample Screenshot 4 :

Sample Screenshot 5 :



JS - Username & Password Validation

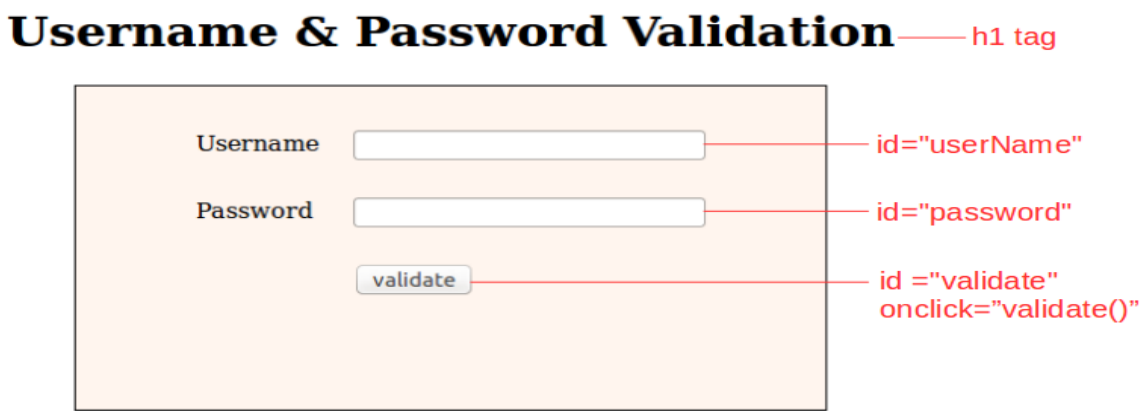
Hints: The test() method tests for a match in a string. This method returns true if it finds a match, otherwise it returns false. Syntax : object.test(string)

Constraints:
The file name should be index.html.
Include an external script file named script.js.
Refer the screenshots for html specifications.

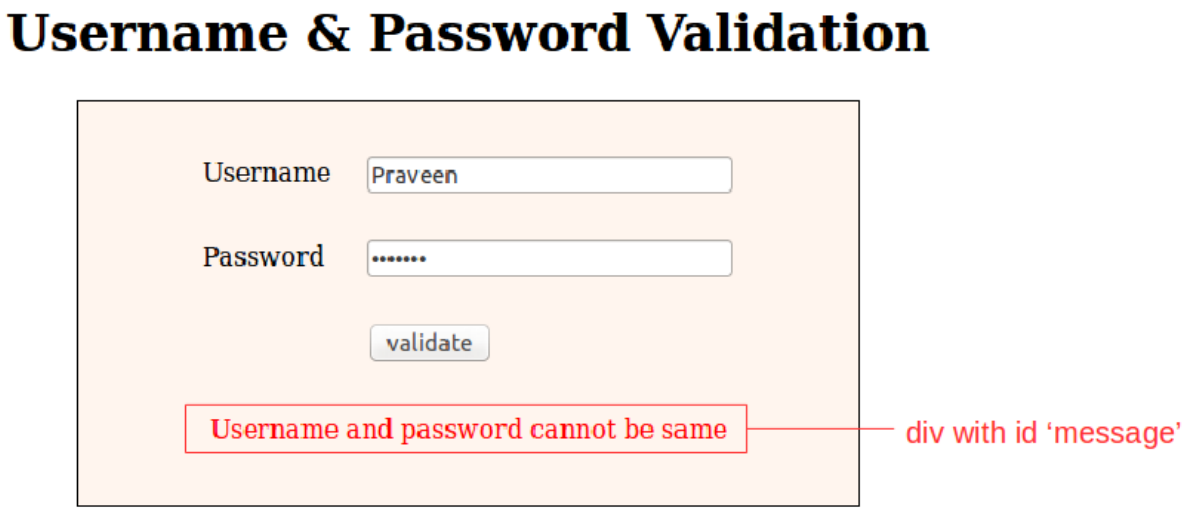
Conditions :
Get the username and password from the user.
Pass the username and password to the javascript function.
Validate the username and password, if it fails, throw a custom exception.
Validation constraints:
1) Username and password must not be same. If they are identical throw 'Username and password cannot be same'.
2) Password must contain atleast 8 characters. If it fails throw 'Password must be minimum of 8 characters'.
3) Password should contain atleast one numberical value, one capital letter and one special character. If it fails throw 'Your password must atleast contain 1 Captial letter, 1 special character, 1 number'.

Note :
Content of the page should be present as shown in the screenshot.
Kindly refer the content which is given as a part of description.

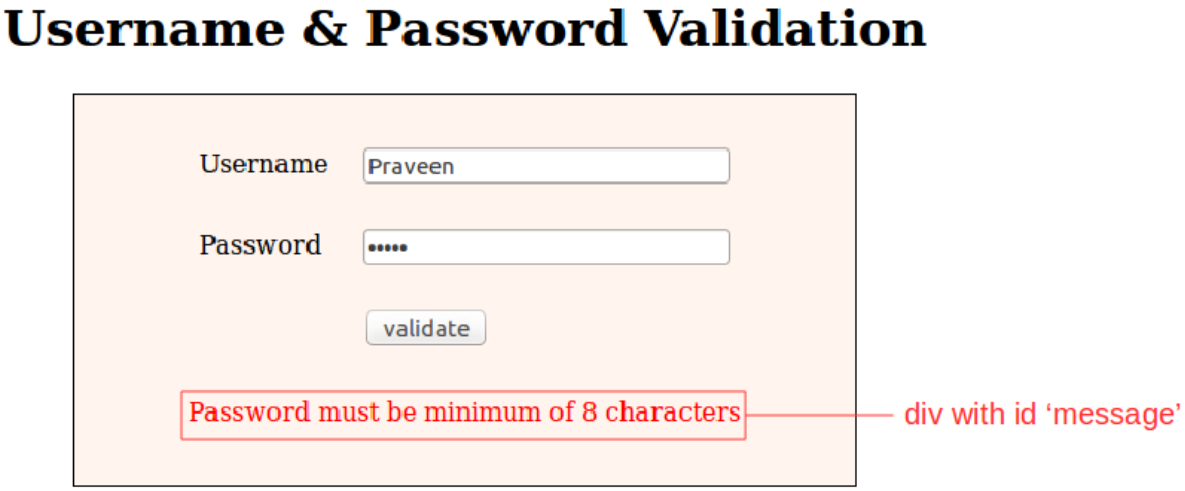
Sample Screenshot 1 :



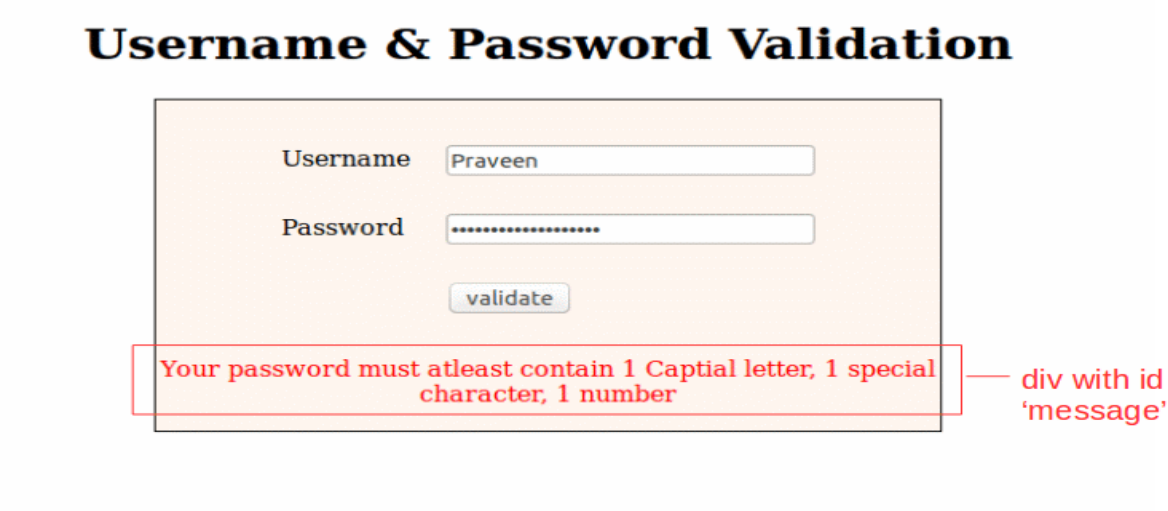
Sample Screenshot 2 :



Sample Screenshot 3 :



Sample Screenshot 4 :



Sample Screenshot 5 :

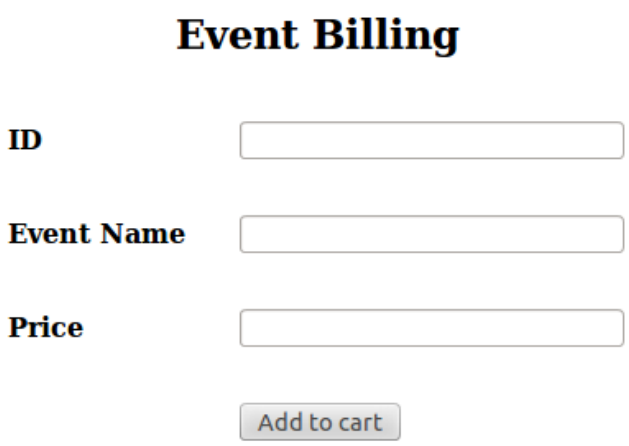
JS - Simple Cart

Constraints :
Design the html page as given in the sample screenshots.
h2 tag - Event Billing
Refer screenshot for id's specification

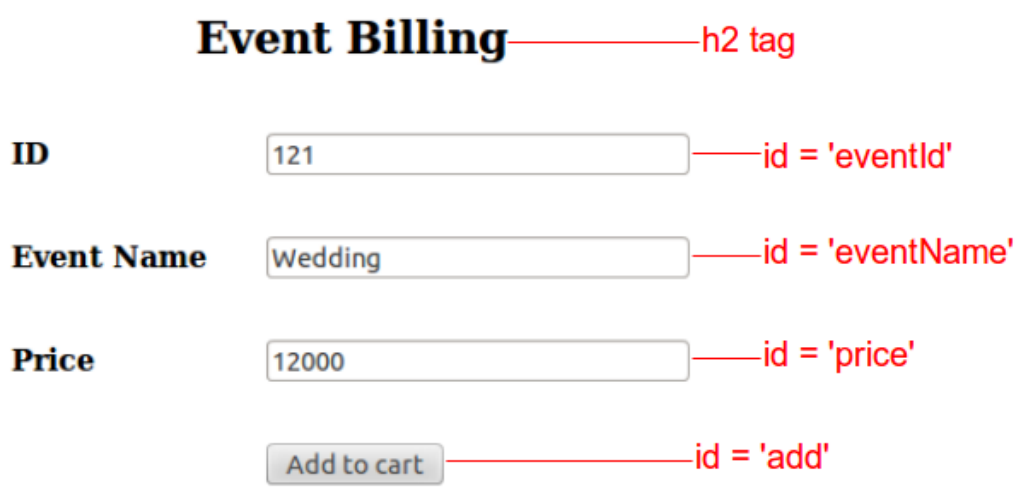
The event details should be stored in an array and the total cost must be displayed along with the event details.
Select box must be present to select the quantities and the total cost should be changed accordingly.
Remove link must be present to delete the event and the total cost should be changed accordingly.
The select box and the remove link should be created dynamically while displaying the details.

Note :
Content of the page should be present as shown in the screenshot.
Kindly refer the content which is given as a part of description.

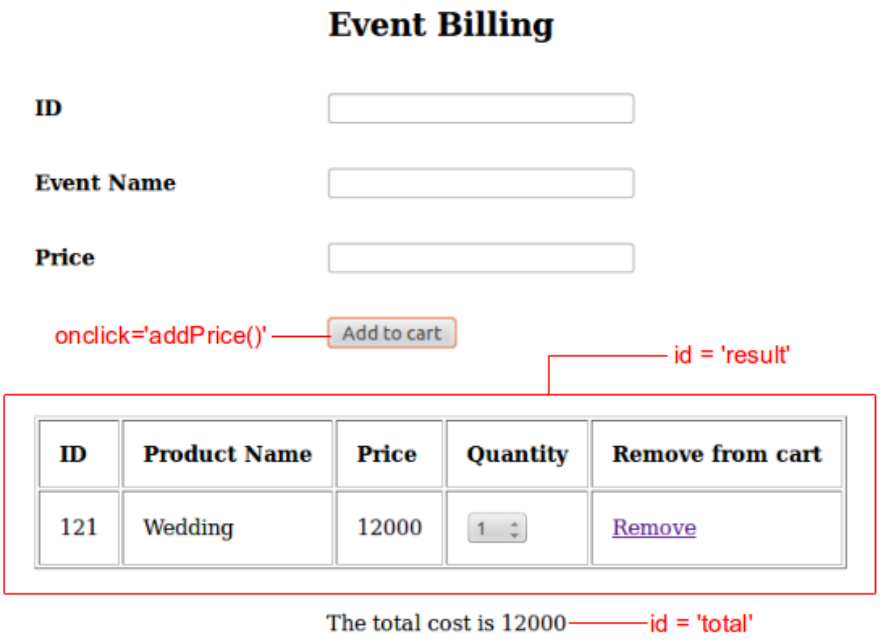
Sample Screenshot 1 :



Sample Screenshot 2 :

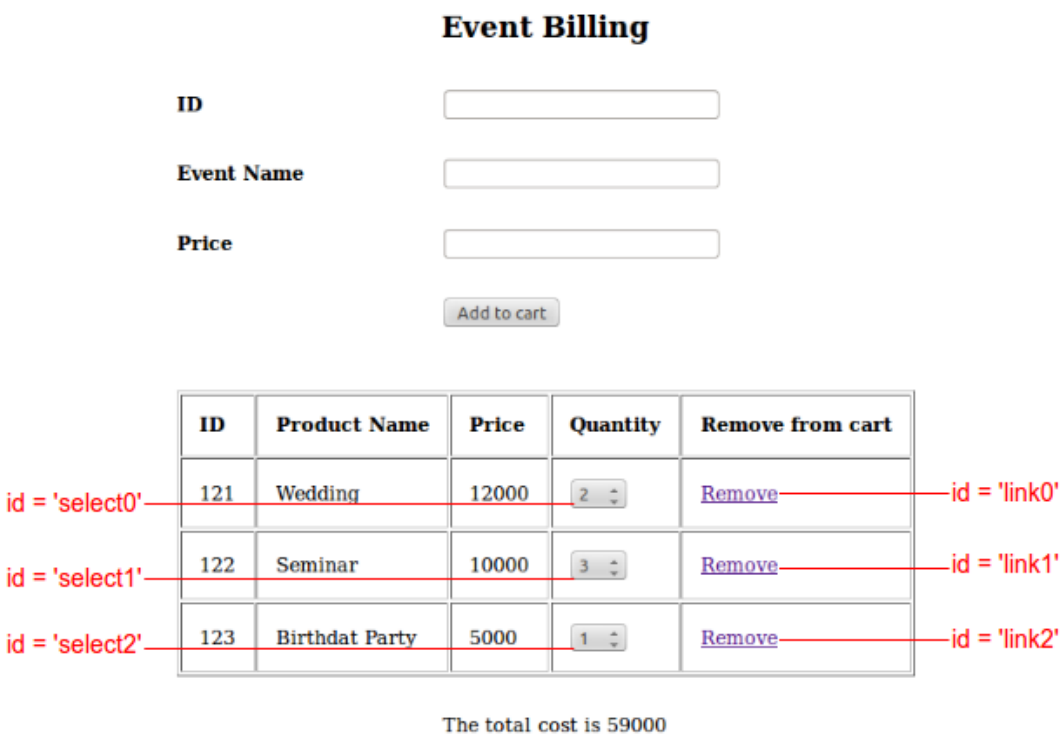


Sample Screenshot 3 :

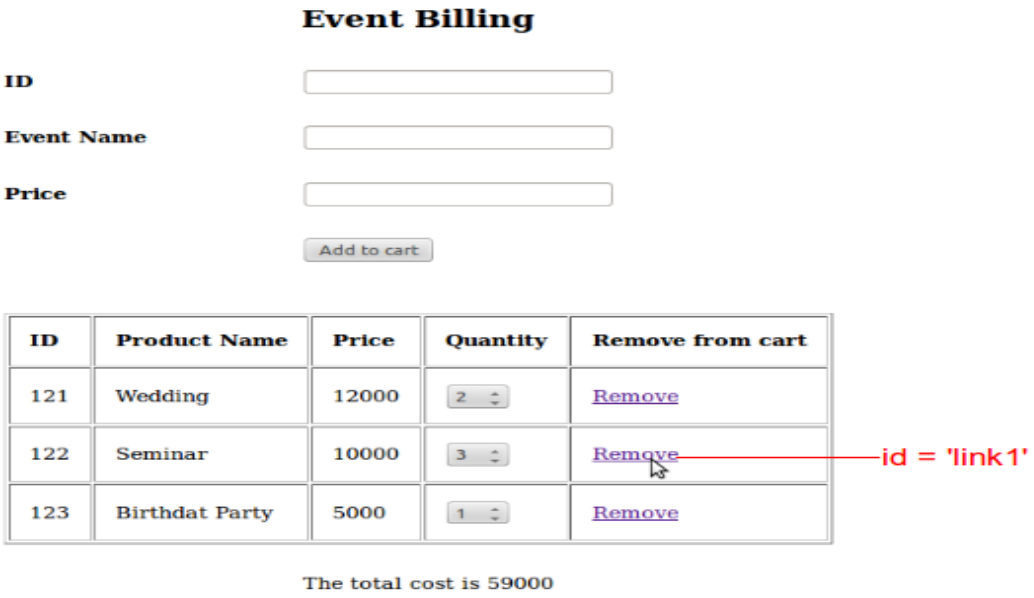


Sample Screenshot 4 :

Dynamically create and assign the id value for remove link (for eg : link0, link1, link2, etc.,)
Dynamically create and assign the id value for select box in quantity column (for eg : select0, select1, select2, etc.,)



Sample Screenshot 5 :



Sample Screenshot 6 :

Event Billing

ID

Event Name

Price

Add to cart

ID	Product Name	Price	Quantity	Remove from cart
121	Wedding	12000	2	Remove
123	Birthday Party	5000	1	Remove

The total cost is 29000