

Experiment -1.1

Student Name: Chayan Gope

Branch: AITCSE-DevOps

Semester: Fifth

Subject Name – Docker & Kubernetes

UID: 22BDO10036

Section/Group: 22BCD-1(A)

Date of Performance: 22/07/2024

Subject Code: 22CSH-343

1. Aim of the practical: To write *JavaScript* to validate the following fields of the Registration page and it should follow the following conditions:

- a) **First Name** - The name should contain alphabets and the length should not be less than 6 characters.
- b) **Password** - The password should not be less than 6 characters in length.
- c) **E-mail ID** - Should not contain any invalid and must follow the standard pattern (for example xyz@gmail.com)
- d) **Mobile Number** - The phone number should contain 10 digits only.
- e) **Last Name and Address** - Should not be Empty.

2. Task to be done: To write JavaScript code that validates the following fields on a Registration page to ensure that the inputs are correctly formatted and meet the specified requirements.

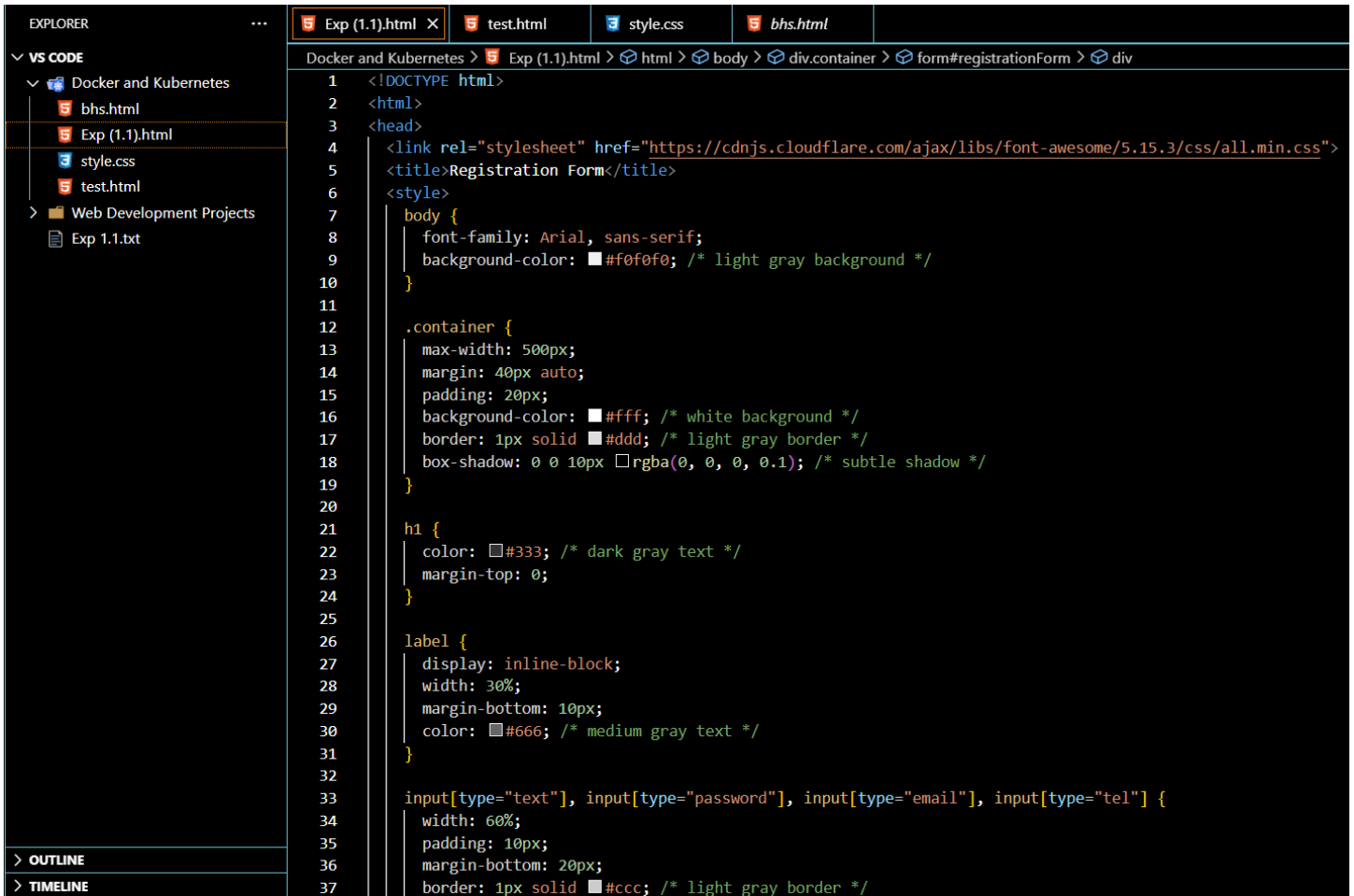
3. Software Used: VS Code, Google Chrome

4. Languages Used: HTML, CSS & JavaScript

5. Steps for experiment:

1. Create an HTML form with fields for First Name, Last Name, Password, E-mail ID, Mobile Number, and Address.
2. Add basic HTML structure including ``<html>``, ``<head>``, and ``<body>`` tags.
3. Create input fields for First Name, Last Name, Password, E-mail ID, Mobile Number, and Address.
4. Include labels and placeholders for each input field for better user understanding.
5. Use CSS to style the form, ensuring it is visually appealing and user-friendly. Here we have created scripts for styling.
6. Add classes and IDs for each element to facilitate easy styling and JavaScript manipulation.
7. Write a JavaScript function to validate
 - a. First Name: Ensure the input contains only alphabets & check if the length is at least 6 characters.
 - b. Password: Ensure the input length is at least 6 characters & ensure it contains at least 1 special symbol(no spaces) and 1 alphabet in capital.
 - c. E-mail ID: Ensure the input follows the pattern (xyz@domain.com)
 - d. Mobile Number: Ensure the input contains exactly 10 digits and should start with the digits 6 7 or 8 or 9
 - e. Last Name and Address: Ensure both fields are not empty.
8. Attach JavaScript validation functions to the form fields ``input` or `submit` events.`
9. Create elements to display error messages next to the respective input fields.
10. Update the error message elements based on validation results.
11. Prevent form submission if any field fails validation.
12. Display appropriate error messages for each invalid field.
13. Test the form to ensure all validations are working correctly.

7. Observations/Discussions:



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
5    <title>Registration Form</title>
6  </head>
7  <body>
8    <div class="container">
9      <div class="row">
10        <div class="col-md-12">
11          <h1>Registration Form</h1>
12          <div class="form">
13            <div class="row">
14              <div class="col-md-6">
15                <input type="text">
16              </div>
17              <div class="col-md-6">
18                <input type="password">
19              </div>
20            </div>
21            <div class="row">
22              <div class="col-md-6">
23                <input type="email">
24              </div>
25              <div class="col-md-6">
26                <input type="tel">
27              </div>
28            </div>
29            <div class="row">
30              <div class="col-md-12">
31                <input type="submit" value="Register">
32              </div>
33            </div>
34          </div>
35        </div>
36      </div>
37    </div>

```

Basic HTML structure including ``<html>``, ``<head>``, and ``<body>`` tags.

EXPLORER

VS CODE

Docker and Kubernetes

bhs.html

Exp (1.1).html

style.css

test.html

Web Development Projects

Exp 1.1.txt

Exp (1.1).html

test.html

style.css

bhs.html

Docker and Kubernetes

Exp (1.1).html

html

body

div.container

form#registrationForm

div

2 <html>

3 <head>

6 <style>

40 input[type="submit"] {

41 background-color: #4CAF50; /* green background */

42 color: #fff; /* white text */

43 padding: 10px 20px;

44 border: none;

45 border-radius: 5px;

46 cursor: pointer;

47 }

48

49 input[type="submit"]:hover {

50 background-color: #3e8e41; /* darker green background */

51 }

52

53 .error {

54 color: red; /* red text */

55 font-size: 14px;

56 margin-bottom: 10px;

57 }

58

59 #errorMessages {

60 padding: 10px;

61 background-color: #ffeaea; /* light pink background */

62 border: 1px solid #ffcccc; /* light pink border */

63 }

64 .eye-icon {

65 position: absolute;

66 right: 10px;

67 top: 50%;

68 transform: translateY(-50%);

69 cursor: pointer;

70 }

71

72 #eye-icon {

73 font-size: 18px;

OUTLINE

TIMELINE

Styling our form

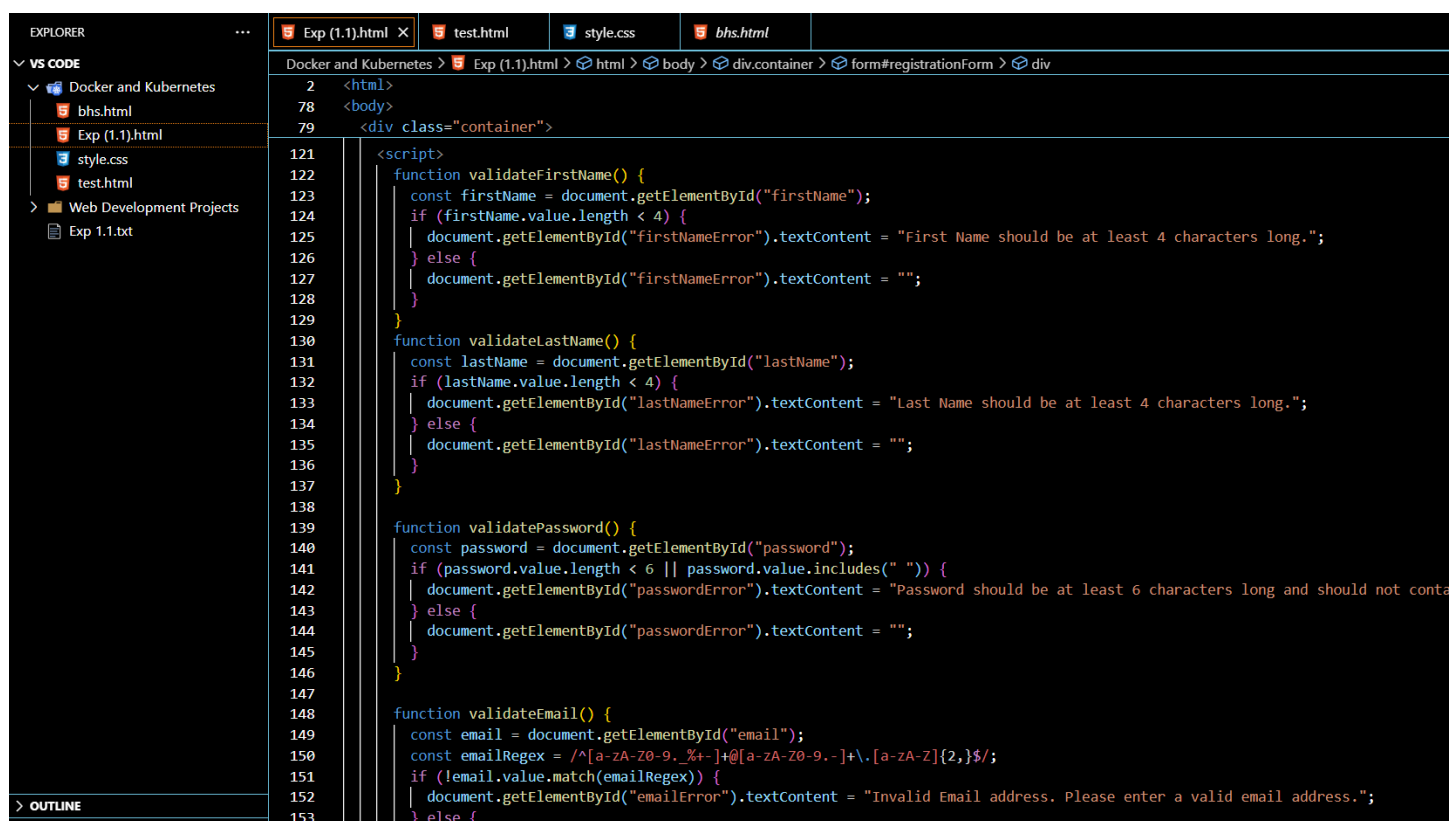
```

EXPLORED
VS CODE
  Docker and Kubernetes
    bhs.html
    Exp (1.1).html
    style.css
    test.html
  Web Development Projects
    Exp 1.1.txt

Exp (1.1).html X test.html style.css bhs.html
Docker and Kubernetes > Exp (1.1).html > html > body > div.container > form#registrationForm > div
2 <html>
78 <body>
79 <div class="container">
80 <h1 align="center"> Registration Form </h1>
81 <h4 align="center"> Fill the form to create your account </h4>
82 <form id="registrationForm">
83 <div>
84 <label for="firstName">First Name:</label>
85 <input type="text" id="firstName" name="firstName" oninput="validateFirstName()">
86 <span id="firstNameError" class="error"></span>
87 </div>
88 <div>
89 <label for="lastName">Last Name:</label>
90 <input type="text" id="lastName" name="lastName" oninput="validateLastName()">
91 <span id="lastNameError" class="error"></span>
92 </div>
93 <div>
94 <label for="email">Email:</label>
95 <input type="email" id="email" name="email" oninput="validateEmail()">
96 <span id="emailError" class="error"></span>
97 </div>
98 <div>
99 <label for="password">Password:</label>
100 <input type="password" id="password" placeholder="Password">
101 <span class="eye-icon" onclick="togglePasswordVisibility()">
102 <i class="fas fa-eye" id="eye-icon"></i>
103 </span>
104 <i id="passwordEye" class="fa fa-eye eye-icon" aria-hidden="true" onclick="togglePasswordVisibility()"></i>
105 </div>
106 <div>
107 <label for="address">Address:</label>
108 <textarea id="address" name="address" oninput="validateAddress()"></textarea>
109 <span id="addressError" class="error"></span>
110 </div>
111 <div>
112 <label for="mobileNumber">Mobile Number:</label>
113 <input type="tel" id="mobileNumber" name="mobileNumber" oninput="validateMobileNumber()">

```

Creating input fields for First Name, Last Name, Password, E-mail ID, Mobile Number, and Address.



```

2 <html>
78 <body>
79 <div class="container">
121 <script>
122   function validateFirstName() {
123     const firstName = document.getElementById("firstName");
124     if (firstName.value.length < 4) {
125       document.getElementById("firstNameError").textContent = "First Name should be at least 4 characters long.";
126     } else {
127       document.getElementById("firstNameError").textContent = "";
128     }
129   }
130   function validateLastName() {
131     const lastName = document.getElementById("lastName");
132     if (lastName.value.length < 4) {
133       document.getElementById("lastNameError").textContent = "Last Name should be at least 4 characters long.";
134     } else {
135       document.getElementById("lastNameError").textContent = "";
136     }
137   }
138   function validatePassword() {
139     const password = document.getElementById("password");
140     if (password.value.length < 6 || password.value.includes(" ")) {
141       document.getElementById("passwordError").textContent = "Password should be at least 6 characters long and should not contain spaces.";
142     } else {
143       document.getElementById("passwordError").textContent = "";
144     }
145   }
146   function validateEmail() {
147     const email = document.getElementById("email");
148     const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
149     if (!email.value.match(emailRegex)) {
150       document.getElementById("emailError").textContent = "Invalid Email address. Please enter a valid email address.";
151     } else {
152       document.getElementById("emailError").textContent = "";
153     }
154   }
155 }
  
```

Writing a JavaScript function to validate

EXPLORER

VS CODE

Docker and Kubernetes

bhs.html

Exp (1.1).html

style.css

test.html

Web Development Projects

Exp 1.1.txt

Exp (1.1).html

test.html

style.css

bhs.html

Docker and Kubernetes > Exp (1.1).html > html > body > div.container > form#registrationForm > div

2 <html>

78 <body>

79 <div class="container">

121 <script>

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

```

function validateMobileNumber() {
  const mobileNumber = document.getElementById("mobileNumber");
  if (mobileNumber.value.length !== 10) {
    document.getElementById("mobileNumberError").textContent = "Mobile Number should be 10 digits long.";
  } else {
    document.getElementById("mobileNumberError").textContent = "";
  }
}


function validateAddress() {
  const address = document.getElementById("address");
  if (address.value.trim() === "") {
    document.getElementById("addressError").textContent = "Address cannot be empty.";
  } else {
    document.getElementById("addressError").textContent = "";
  }
}

function submitForm() {
  const errorMessages = document.getElementById("errorMessages");
  const errors = [];

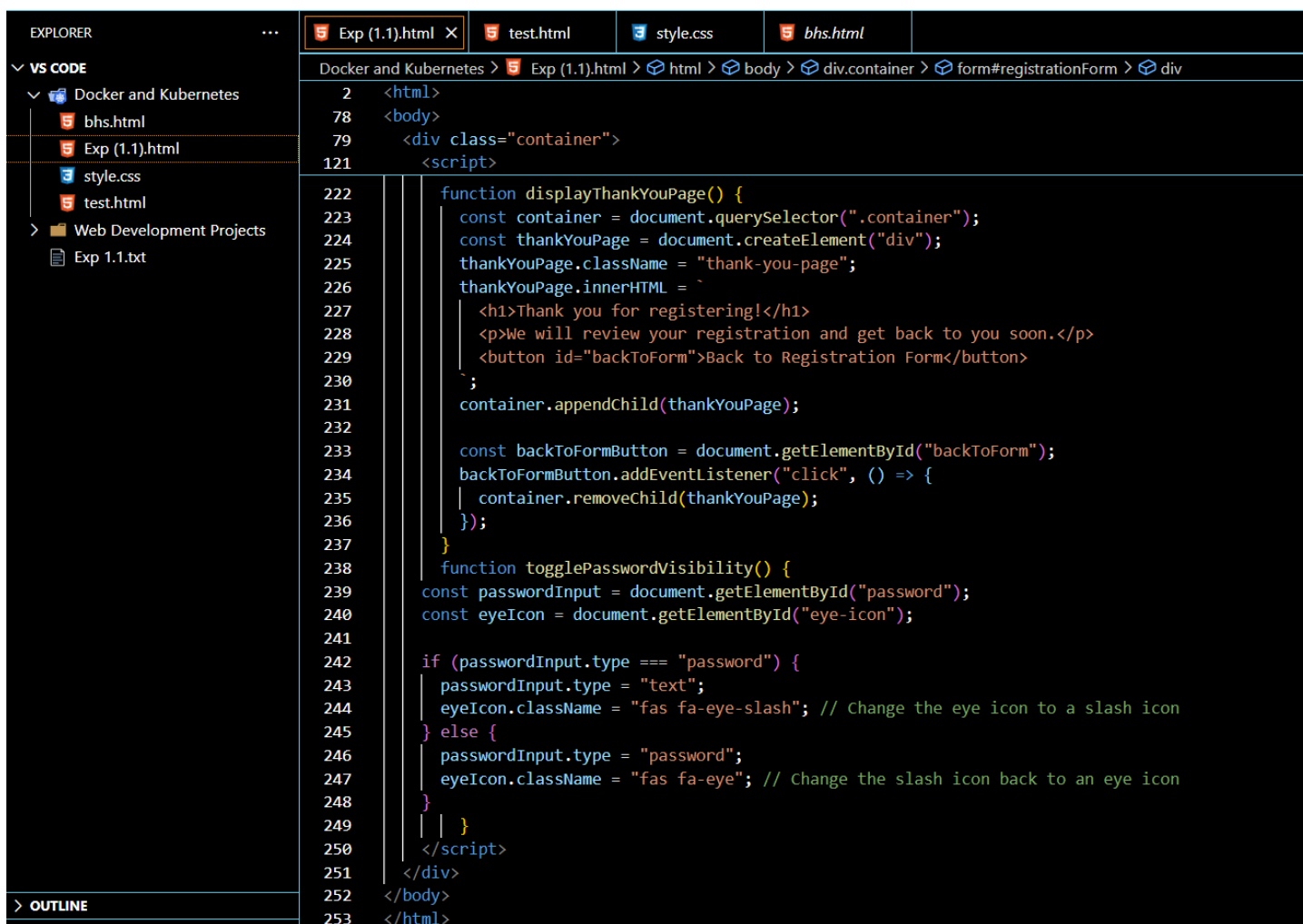
  const firstNameError = document.getElementById("firstNameError").textContent;
  const lastNameError = document.getElementById("lastNameError").textContent;
  const passwordError = document.getElementById("passwordError").textContent;
  const emailError = document.getElementById("emailError").textContent;
  const mobileNumberError = document.getElementById("mobileNumberError").textContent;
  const addressError = document.getElementById("addressError").textContent;

  if (firstNameError !== "") {
    errors.push(firstNameError);
  }
}

```



✉ egov@cumail.in



```

2 <html>
78 <body>
79 <div class="container">
121 <script>
222     function displayThankYouPage() {
223         const container = document.querySelector(".container");
224         const thankYouPage = document.createElement("div");
225         thankYouPage.className = "thank-you-page";
226         thankYouPage.innerHTML = `
227             <h1>Thank you for registering!</h1>
228             <p>We will review your registration and get back to you soon.</p>
229             <button id="backToForm">Back to Registration Form</button>
230         `;
231         container.appendChild(thankYouPage);
232
233         const backToFormButton = document.getElementById("backToForm");
234         backToFormButton.addEventListener("click", () => {
235             container.removeChild(thankYouPage);
236         });
237     }
238     function togglePasswordVisibility() {
239         const passwordInput = document.getElementById("password");
240         const eyeIcon = document.getElementById("eye-icon");
241
242         if (passwordInput.type === "password") {
243             passwordInput.type = "text";
244             eyeIcon.className = "fas fa-eye-slash"; // Change the eye icon to a slash icon
245         } else {
246             passwordInput.type = "password";
247             eyeIcon.className = "fas fa-eye"; // Change the slash icon back to an eye icon
248         }
249     }
250 </script>
251 </div>
252 </body>
253 </html>

```

Creating elements to display error messages next to the respective input fields. Updating the error message elements based on validation results.

Registration Form

Fill the form to create your account

First Name:

Last Name:

Email:

Password:

Address:

Mobile Number:

Register

Our Form Looks like this

Registration Form

Fill the form to create your account

First Name:

abcd

Last Name:

cdef

Email:

abcd@gmail.com

Password:

.....

Address:

Chandigarh

Mobile Number:

7589648355

Register

All entered fields of the form the correct.

10. Result/Output/Writing Summary: The experiment successfully implemented JavaScript validation for the registration page fields

Learning outcomes (What I have learned):

1. Learnt about HTML
2. Learnt about JavaScript.
3. Learnt about CSS.
4. Learnt about validating fields in JS
5. Learnt about creating a registration form.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			