

ETHNUS

MERN STACK

JavaScript Assignment

Yuvraj Singh
22BEY10079
yuvrajsingh2002@gmail.com

TASK - 1

Write a JavaScript function that reverse a number.

Sample Data and output:

Example x = 32243;

Expected Output : 34223

```
Last login: Wed Mar 26 18:00:28 on ttys000
[yuvraj@Yuvrajs-MacBook-Pro ~ % node
Welcome to Node.js v22.13.1.
Type ".help" for more information.
> function reverseNumber(num) {
...     return parseInt(num.toString().split('').reverse().join(''), 10);
... }
undefined
>
> // Example usage:
undefined
> let x = 32243;
undefined
[> console.log(reverseNumber(x));
34223
undefined
> █
```

Code run in Terminal

TASK - 2

- Write a JavaScript function that returns a passed string with letters in alphabetical order.

Example string : 'webmaster'

Expected Output : 'abeemrstw'

```
Last login: Fri Mar 28 00:01:10 on ttys000
[yuvraj@Yuvrajs-MacBook-Pro ~ % node
Welcome to Node.js v22.13.1.
Type ".help" for more information.
> function alphabetizeString(str) {
...     return str.split('').sort().join('');
... }
undefined
>
> // Example usage:
undefined
> let example = 'webmaster';
undefined
[> console.log(alphabetizeString(example)); // Output: 'abeemrstw'
abeemrstw
undefined
> █
```

TASK - 3

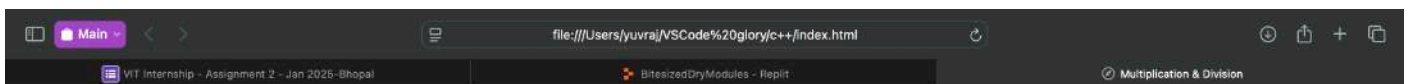
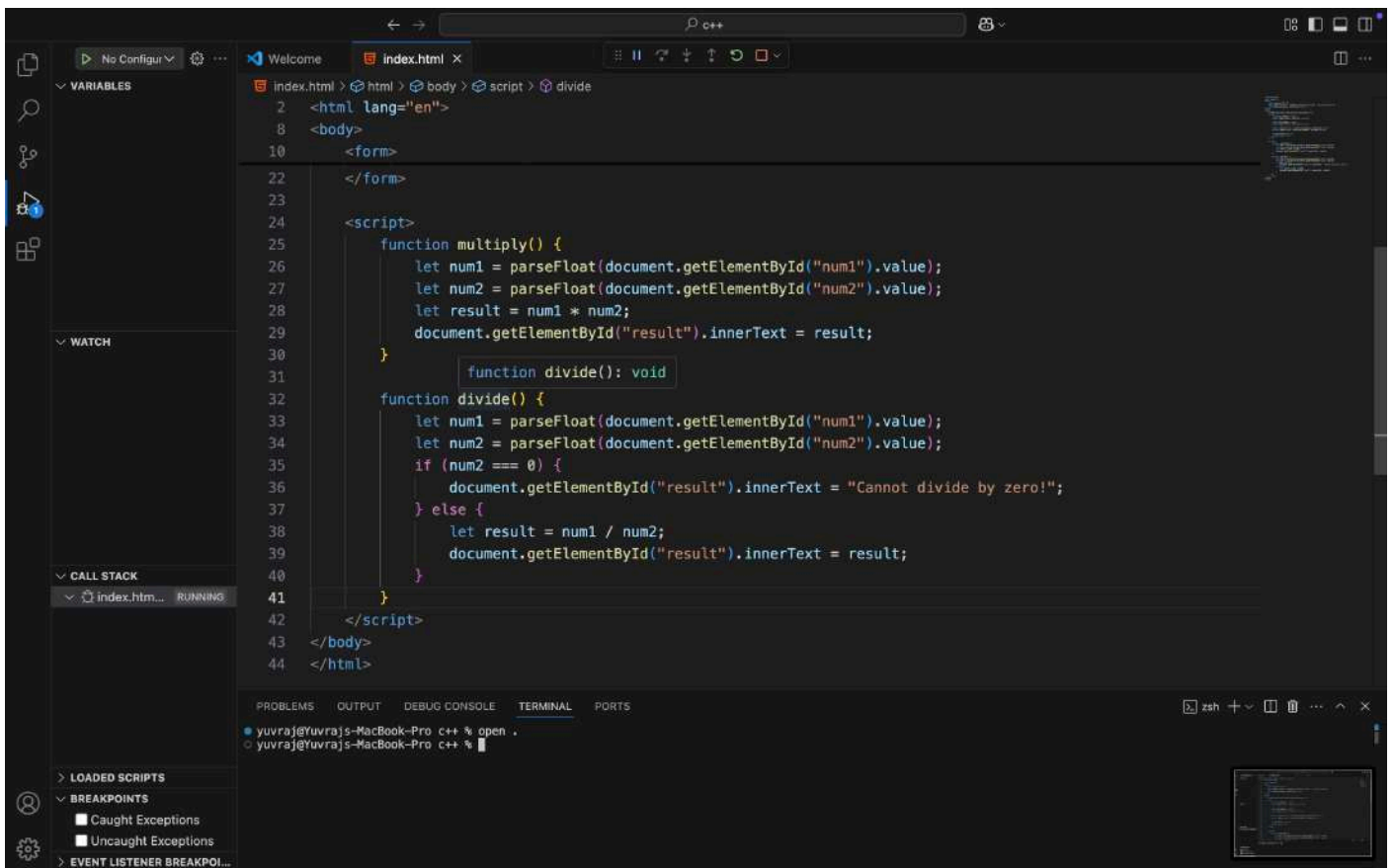
- Write a JavaScript program to calculate multiplication and division of two numbers (input from user).
- Sample Form:

1st Number :

2nd Number:

The Result Is :
120

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Multiplication & Division</title>
7 </head>
8 <body>
9   <h2>Multiplication and Division Calculator</h2>
10  <form>
11    <label>1st Number:</label>
12    <input type="number" id="num1"><br><br>
13
14    <label>2nd Number:</label>
15    <input type="number" id="num2"><br><br>
16
17    <button type="button" onclick="multiply()">Multiply</button>
18    <button type="button" onclick="divide()">Divide</button>
19
20    <h3>The Result Is:</h3>
21    <p id="result"></p>
22  </form>
23
24  <script>
25    function multiply() {
26      let num1 = parseFloat(document.getElementById("num1").value);
27      let num2 = parseFloat(document.getElementById("num2").value);
```



Multiplication and Division Calculator

1st Number:

2nd Number:

The Result Is:

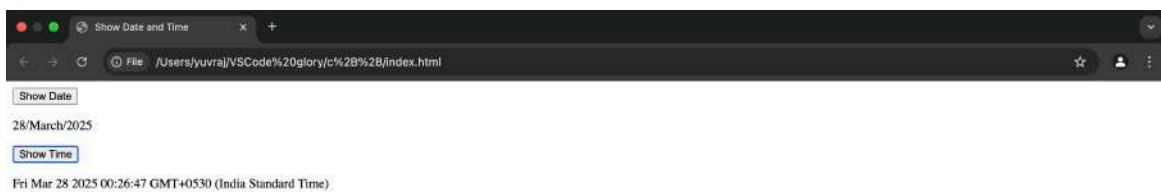
76908

TASK - 4



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Show Date and Time</title>
7 </head>
8 <body>
9
10  <button onclick="showDate()">Show Date</button>
11  <p id="date"></p>
12    function showTime(): void
13  <button onclick="showTime()">Show Time</button>
14  <p id="time"></p>
15
16  <script>
17    function showDate() {
18      let today = new Date();
19      let formattedDate = today.getDate() + "/" +
20        today.toLocaleString('default', { month: 'long' }) + "/" +
21        today.getFullYear();
22      document.getElementById("date").innerText = formattedDate;
23    }
24
25    function showTime() {
26      let currentTime = new Date();
27      document.getElementById("time").innerText = currentTime.toString();
28    }
29  </script>
30 </body>
31 </html>
```

```
2 <html lang="en">
7 </head>
8 <body>
9
10 <button onclick="showDate()">Show Date</button>
11 <p id="date"></p>
12
13 <button onclick="showTime()">Show Time</button>
14 <p id="time"></p>
15
16 <script>
17   function showDate() {
18     let today = new Date();
19     let formattedDate = today.getDate() + "/" +
20       today.toLocaleString('default', { month: 'long' }) + "/" +
21       today.getFullYear();
22     document.getElementById("date").innerText = formattedDate;
23   }
24
25   function showTime() {
26     let currentTime = new Date();
27     document.getElementById("time").innerText = currentTime.toString();
28   }
29 </script>
30
31 </body>
32 </html>
```



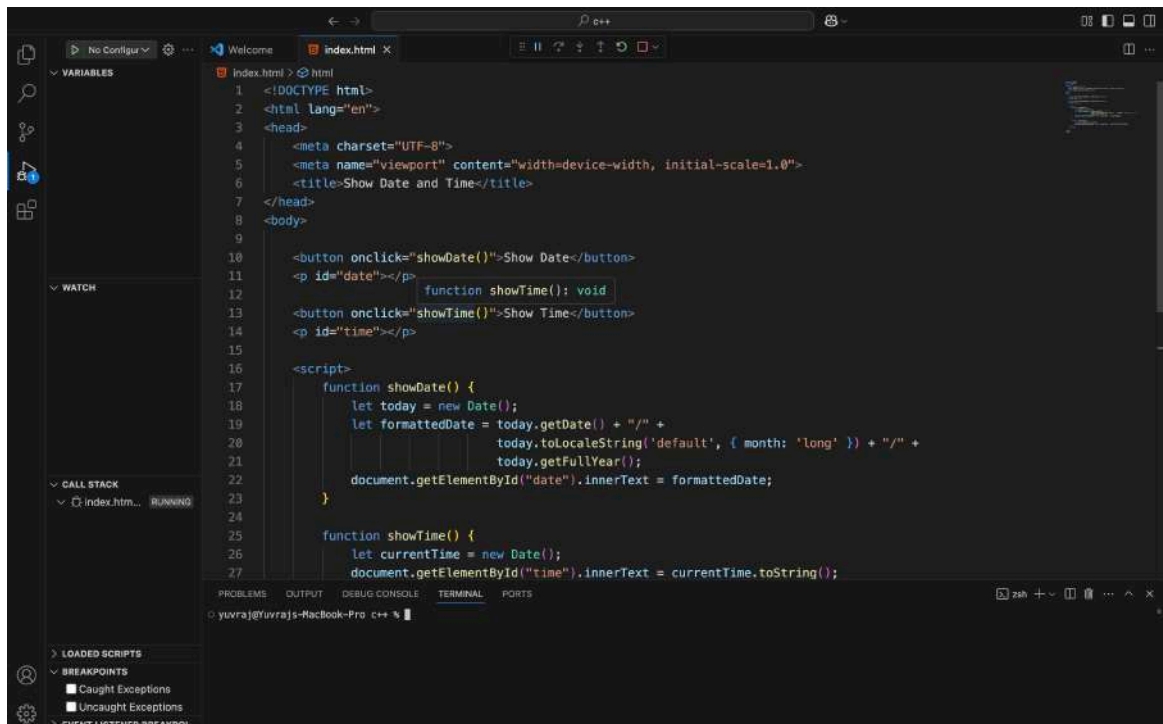
TASK - 5

Name: This field is required

Email: A valid email address is required

Website: A valid url is required

Message: This field is required



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Show Date and Time</title>
7 </head>
8 <body>
9
10  <button onclick="showDate()">Show Date</button>
11  <p id="date"></p>
12    function showTime(): void
13  <button onclick="showTime()">Show Time</button>
14  <p id="time"></p>
15
16  <script>
17    function showDate() {
18      let today = new Date();
19      let formattedDate = today.getDate() + "/" +
20        today.toLocaleString('default', { month: 'long' }) + "/" +
21        today.getFullYear();
22      document.getElementById("date").innerText = formattedDate;
23    }
24
25    function showTime() {
26      let currentTime = new Date();
27      document.getElementById("time").innerText = currentTime.toString();
28    }
29  </script>
30 </body>
31 </html>
```


The screenshot shows the VS Code editor with a file named `index.html` open. The editor is in the "RUN AND DEBUG" mode. The code in `index.html` includes a meta charset of "UTF-8", a viewport meta tag, and a title "Form Validation". It also contains CSS for error messages, with `.error` having a red color and 14px font-size, and `.input-error` having a 2px solid red border. The HTML body contains a form with three input fields: "Name", "Email", and "Website". Each input field has a corresponding error message span with the class "error". The form has an `onsubmit` attribute that calls `return validateForm()`. The terminal at the bottom shows the command `open .` being executed.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Form Validation</title>
7   <style>
8     .error { color: red; font-size: 14px; }
9     .input-error { border: 2px solid red; }
10  </style>
11 </head>
12 <body>
13
14  <form id="myForm" onsubmit="return validateForm()">
15    <label for="name">Name:</label>
16    <input type="text" id="name">
17    <span id="nameError" class="error"></span>
18    <br><br>
19
20    <label for="email">Email:</label>
21    <input type="text" id="email">
22    <span id="emailError" class="error"></span>
23    <br><br>
24
25    <label for="website">Website:</label>
26    <input type="text" id="website">
27    <span id="websiteError" class="error"></span>
```

The screenshot shows the VS Code editor with a file named `index.html` open. The editor is in the "RUN AND DEBUG" mode. The code in `index.html` includes a meta charset of "UTF-8", a viewport meta tag, and a title "Form Validation". It also contains CSS for error messages, with `.error` having a red color and 14px font-size, and `.input-error` having a 2px solid red border. The HTML body contains a form with three input fields: "Name", "Email", and "Website". Each input field has a corresponding error message span with the class "error". The form has an `onsubmit` attribute that calls `return validateForm()`. The terminal at the bottom shows the command `open .` being executed.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Form Validation</title>
7   <style>
8     .error { color: red; font-size: 14px; }
9     .input-error { border: 2px solid red; }
10  </style>
11 </head>
12 <body>
13
14  <form id="myForm" onsubmit="return validateForm()">
15    <label for="name">Name:</label>
16    <input type="text" id="name">
17    <span id="nameError" class="error"></span>
18    <br><br>
19
20    <label for="email">Email:</label>
21    <input type="text" id="email">
22    <span id="emailError" class="error"></span>
23    <br><br>
24
25    <label for="website">Website:</label>
26    <input type="text" id="website">
27    <span id="websiteError" class="error"></span>
28    <br><br>
29
30    <label for="message">Message:</label>
31    <textarea id="message"></textarea>
32    <span id="messageError" class="error"></span>
33    <br><br>
34
35    <button type="submit">Submit</button>
36  </form>
37
38  <script>
39    function validateForm() {
40      let isValid = true;
41
42      // Get input values
43      let name = document.getElementById("name").value.trim();
44      let email = document.getElementById("email").value.trim();
45      let website = document.getElementById("website").value.trim();
46      let message = document.getElementById("message").value.trim();
47
48      // Basic validation logic
```

The image shows the Visual Studio Code editor with a file named `index.html` open. The code is a JavaScript function `validateForm()` that performs form validation. It includes comments for 'Name validation' and 'Email validation (basic regex)'. The email validation uses a regular expression `let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;`. The function checks if the email is empty or matches the pattern, setting `isValid` to `false` and adding an `input-error` class to the email field if it fails. The interface also shows the 'RUN AND DEBUG' sidebar on the left and a terminal at the bottom.

```
2 <html lang="en">
12 <body>
38 <script>
39   function validateForm() {
40
41     // Reset previous errors
42     document.querySelectorAll(".error").forEach(el => el.innerText = "");
43     document.querySelectorAll("input, textarea").forEach(el => el.classList.remove("input-error"));
44
45     // Name validation
46     if (name === "") {
47       document.getElementById("nameError").innerText = "This field is required";
48       document.getElementById("name").classList.add("input-error");
49       isValid = false;
50     }
51
52     // Email validation (basic regex)
53     let emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
54     if (email === "") {
55       document.getElementById("emailError").innerText = "A valid email address is required";
56       document.getElementById("email").classList.add("input-error");
57       isValid = false;
58     } else if (!emailPattern.test(email)) {
59       document.getElementById("emailError").innerText = "Invalid email format";
60       document.getElementById("email").classList.add("input-error");
61       isValid = false;
62     }
63   }
64 }
```

The image shows a web browser window displaying a form titled "VIT Internship - Assignment 2 - Jan 2025-Bhopal". The form has four input fields: "Name", "Email", "Website", and "Message". The "Name" field contains "Yuvraj Singh". The "Email" field contains "yuvraj@" and has a red border with the text "Invalid email format" next to it. The "Website" field contains "svvsdvs" and has a red border with the text "Invalid URL format" next to it. The "Message" field contains "vsdvsdvs". A "Submit" button is at the bottom of the form.

Name: Yuvraj Singh

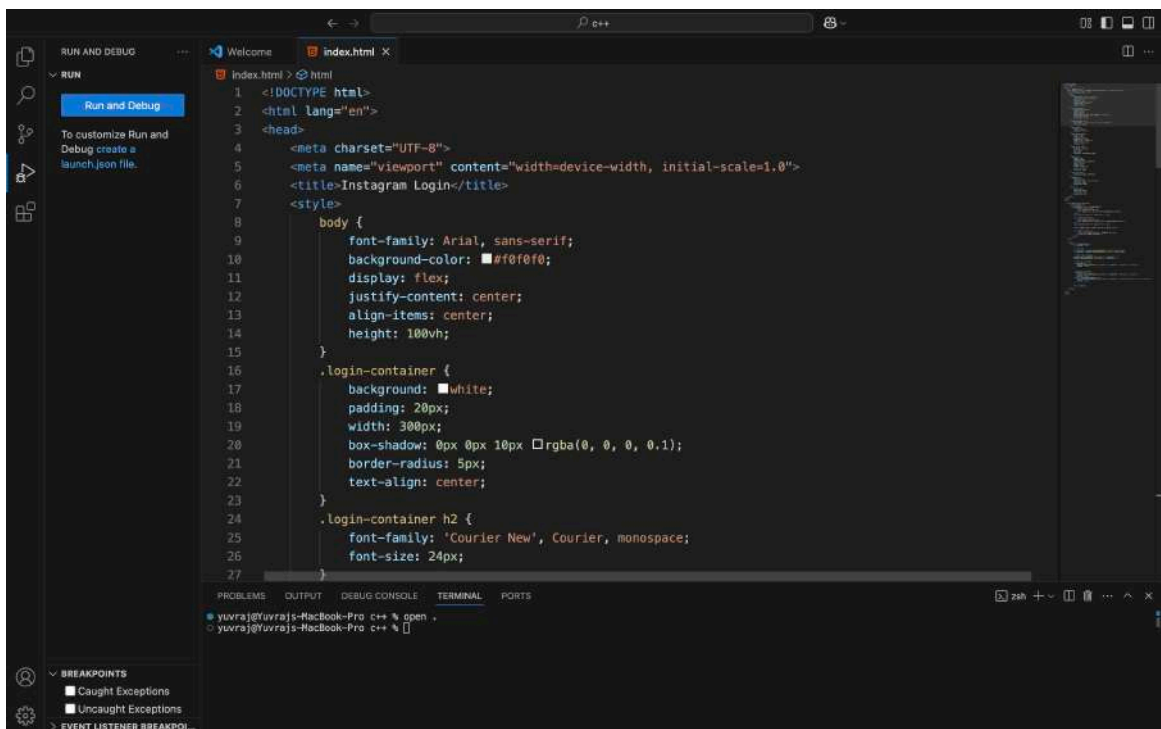
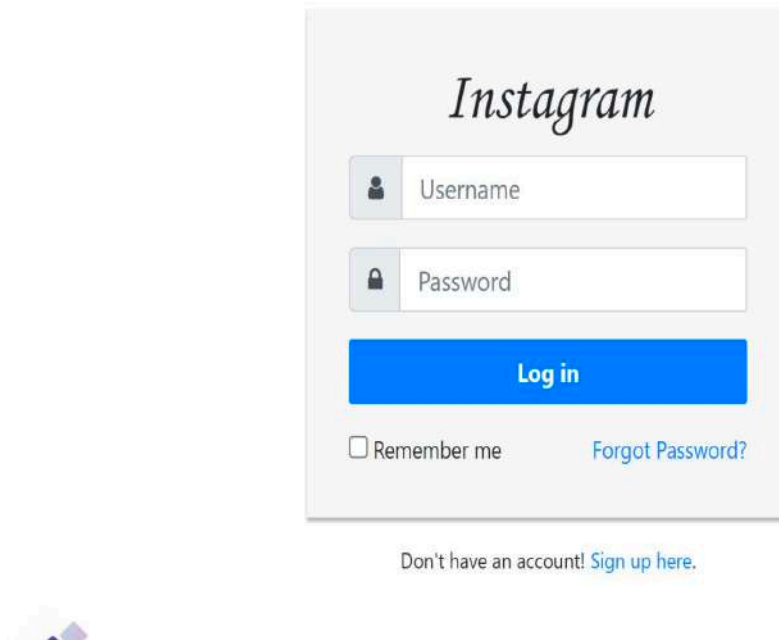
Email: yuvraj@ Invalid email format

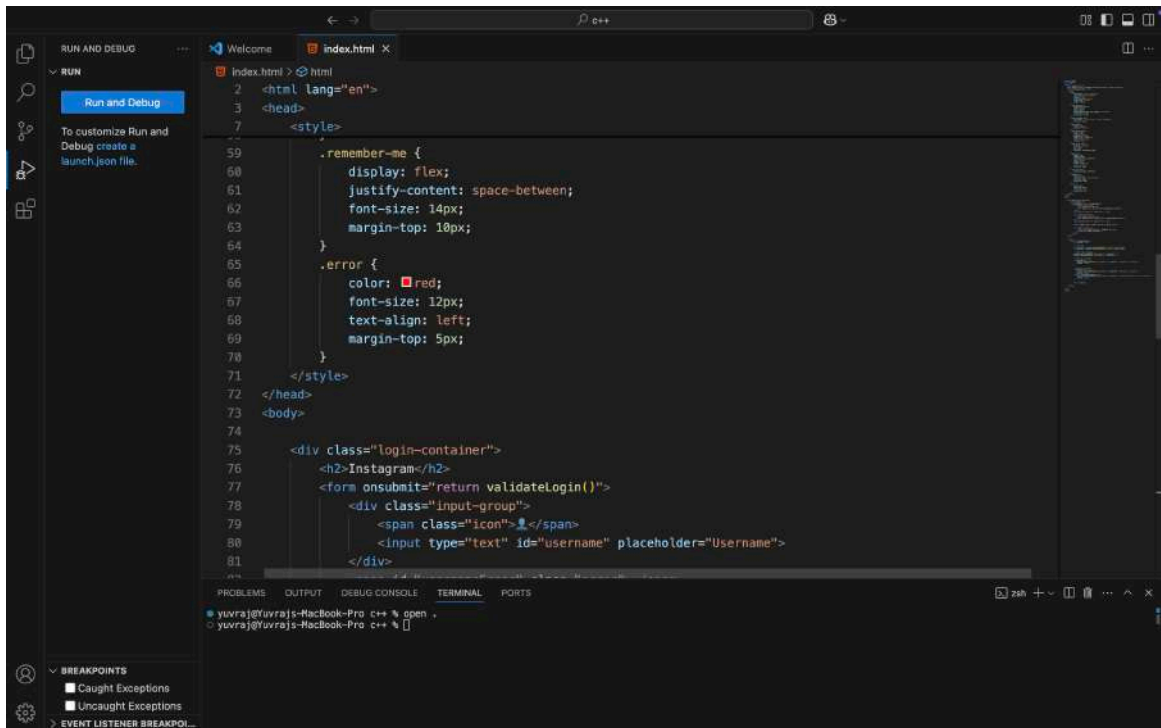
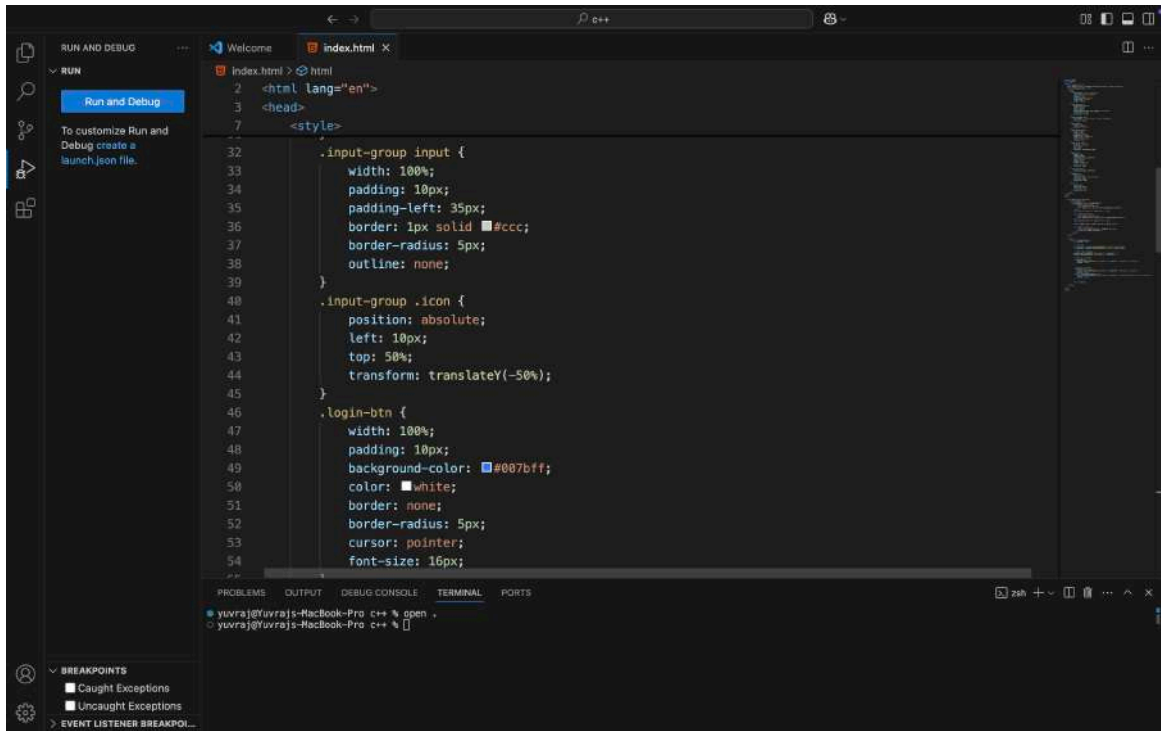
Website: svvsdvs Invalid URL format

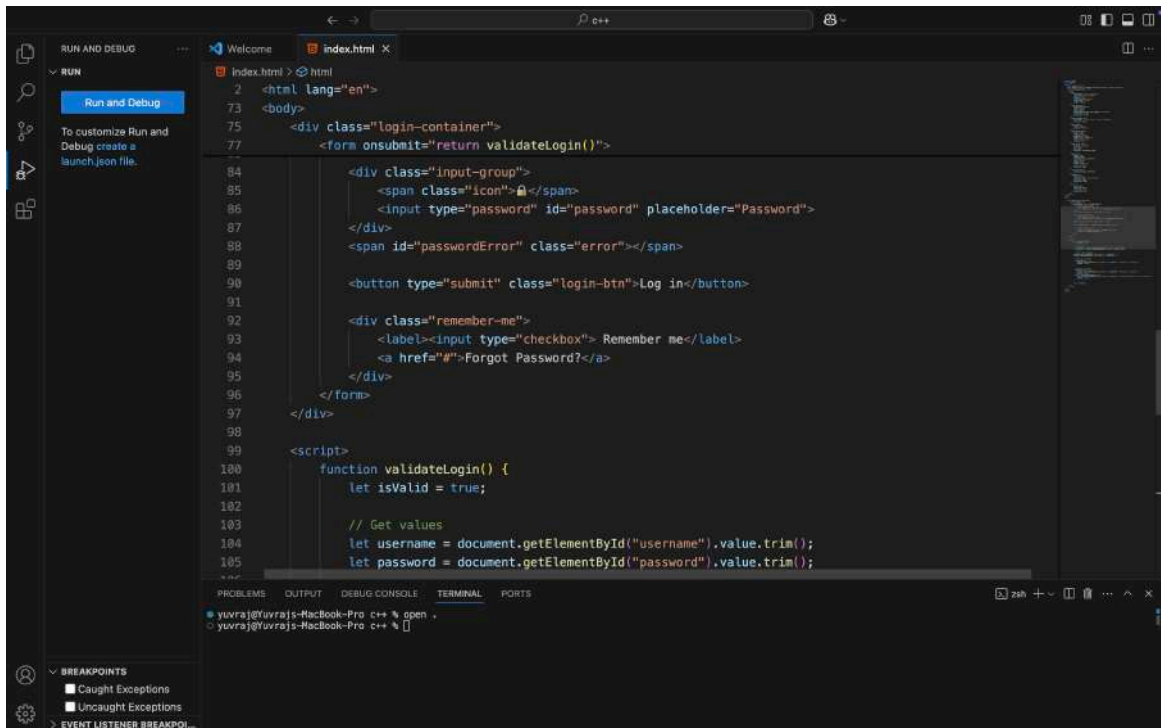
Message: vsdvsdvs

Submit

TASK - 6



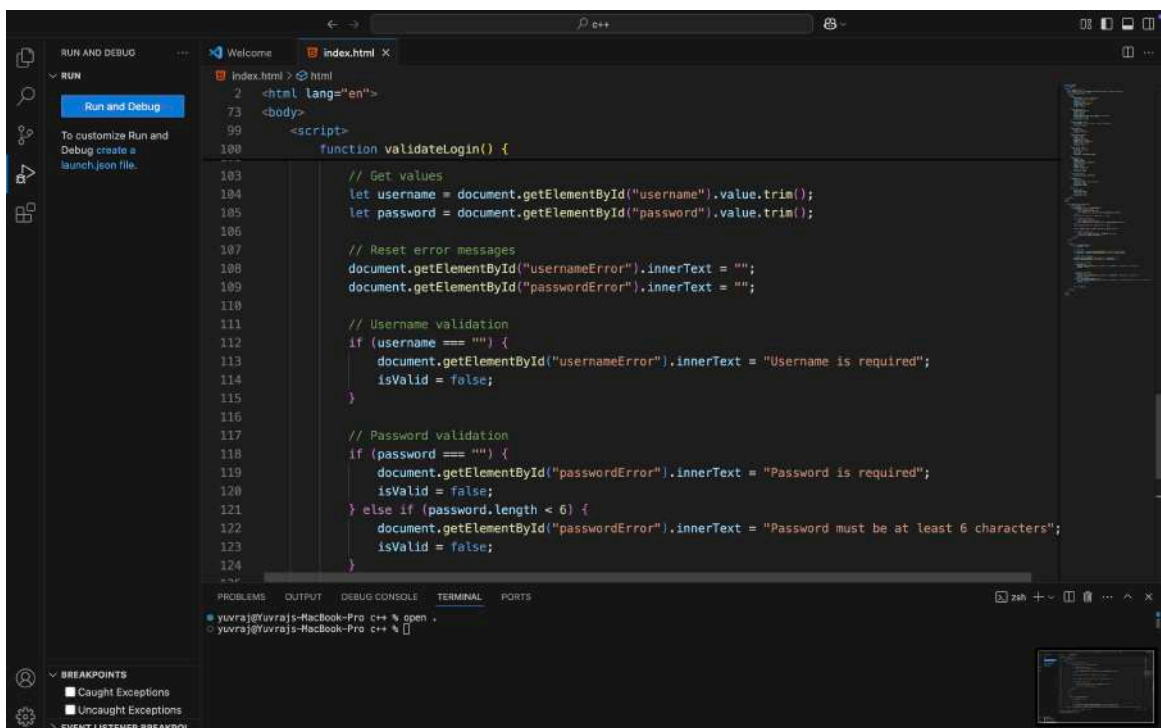




The screenshot shows the Visual Studio Code editor with a file named `index.html` open. The code is written in HTML and JavaScript. The HTML part defines a login form with a password input field, a submit button, and a 'Remember me' checkbox. The JavaScript part defines a `validateLogin()` function that retrieves the username and password values.

```
2 <html lang="en">
73 <body>
75   <div class="login-container">
77     <form onsubmit="return validateLogin()">
84       <div class="input-group">
85         <span class="icon">✉</span>
86         <input type="password" id="password" placeholder="Password">
87       </div>
88       <span id="passwordError" class="error"></span>
89
90       <button type="submit" class="login-btn">Log in</button>
91
92       <div class="remember-me">
93         <label><input type="checkbox"> Remember me</label>
94         <a href="#">Forgot Password?</a>
95       </div>
96     </form>
97   </div>
99   <script>
100     function validateLogin() {
101       let isValid = true;
102
103       // Get values
104       let username = document.getElementById("username").value.trim();
105       let password = document.getElementById("password").value.trim();
106     }
107   </script>
```

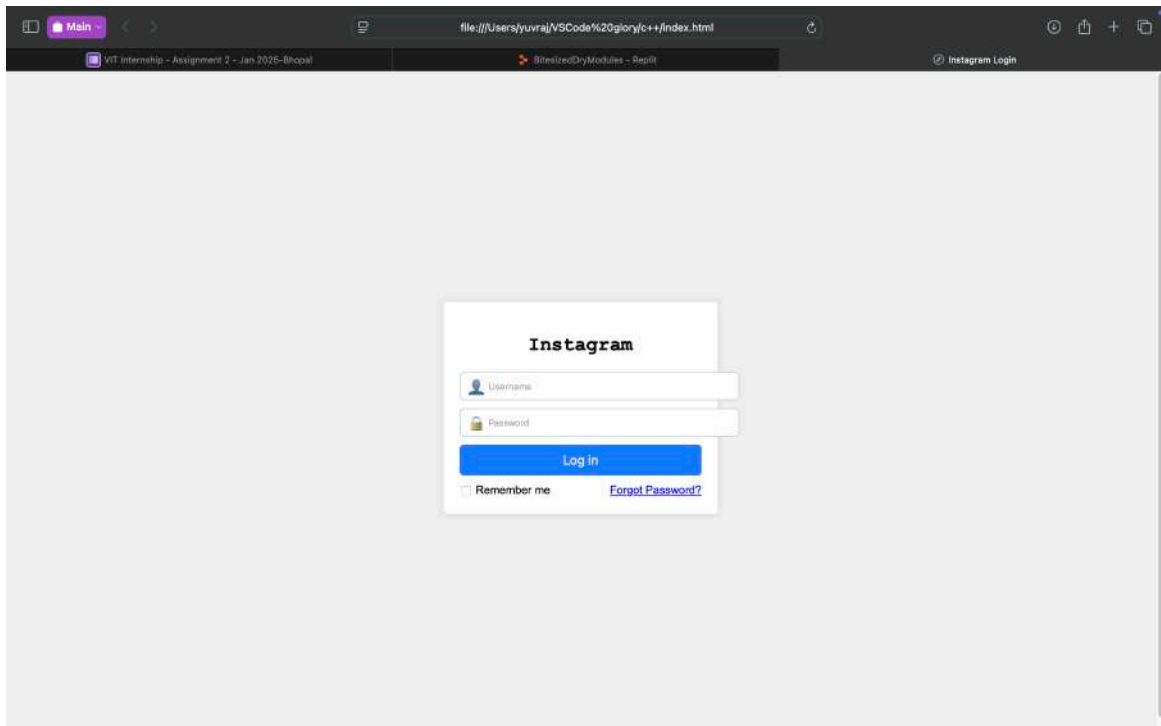
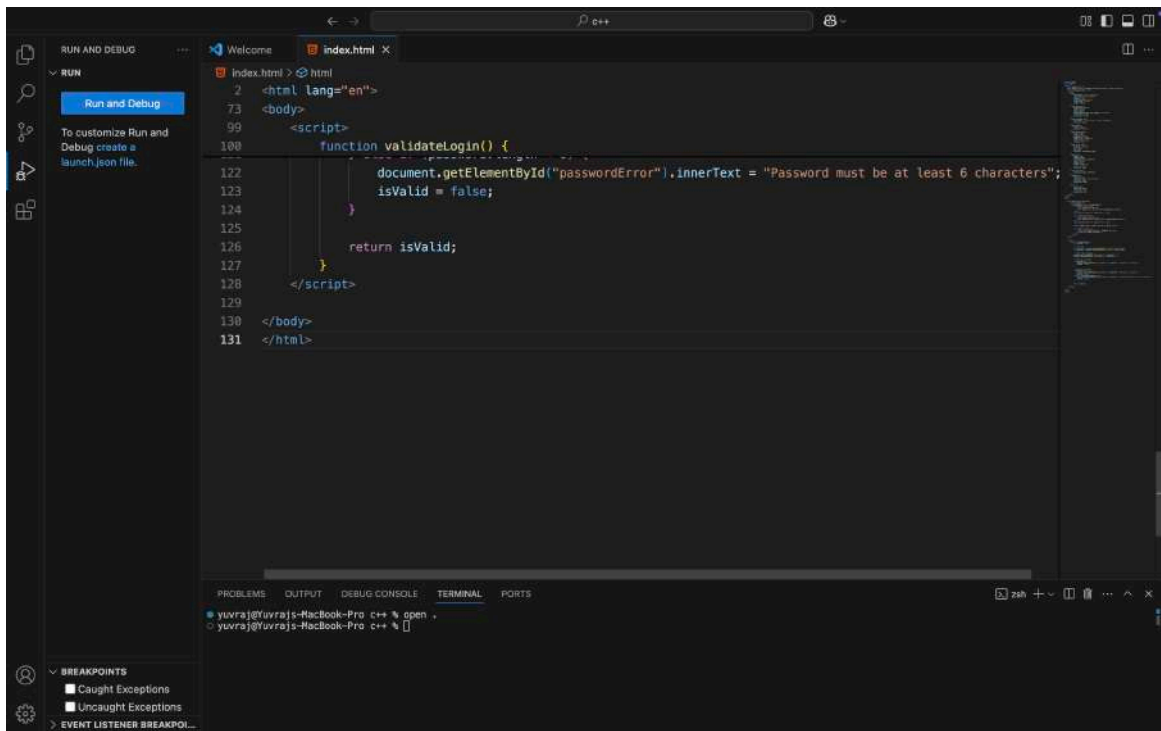
The terminal at the bottom shows the command `yuvraj@Yuvrajs-MacBook-Pro c++ % open .` and the output `yuvraj@Yuvrajs-MacBook-Pro c++ %`.



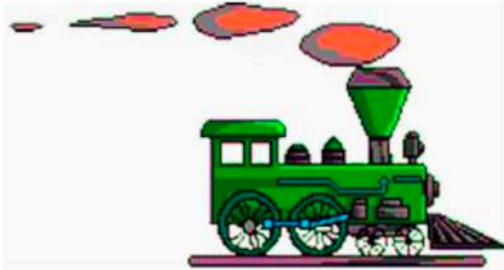
The screenshot shows the Visual Studio Code editor with the same `index.html` file open. The JavaScript code has been updated to include validation logic for the username and password fields. The `validateLogin()` function now checks if the username is empty and if the password is empty or less than 6 characters long. It also resets the error messages for both fields.

```
99   <script>
100     function validateLogin() {
103       // Get values
104       let username = document.getElementById("username").value.trim();
105       let password = document.getElementById("password").value.trim();
106
107       // Reset error messages
108       document.getElementById("usernameError").innerText = "";
109       document.getElementById("passwordError").innerText = "";
110
111       // Username validation
112       if (username === "") {
113         document.getElementById("usernameError").innerText = "Username is required";
114         isValid = false;
115       }
116
117       // Password validation
118       if (password === "") {
119         document.getElementById("passwordError").innerText = "Password is required";
120         isValid = false;
121       } else if (password.length < 6) {
122         document.getElementById("passwordError").innerText = "Password must be at least 6 characters";
123         isValid = false;
124       }
125     }
126   </script>
```

The terminal at the bottom shows the command `yuvraj@Yuvrajs-MacBook-Pro c++ % open .` and the output `yuvraj@Yuvrajs-MacBook-Pro c++ %`.

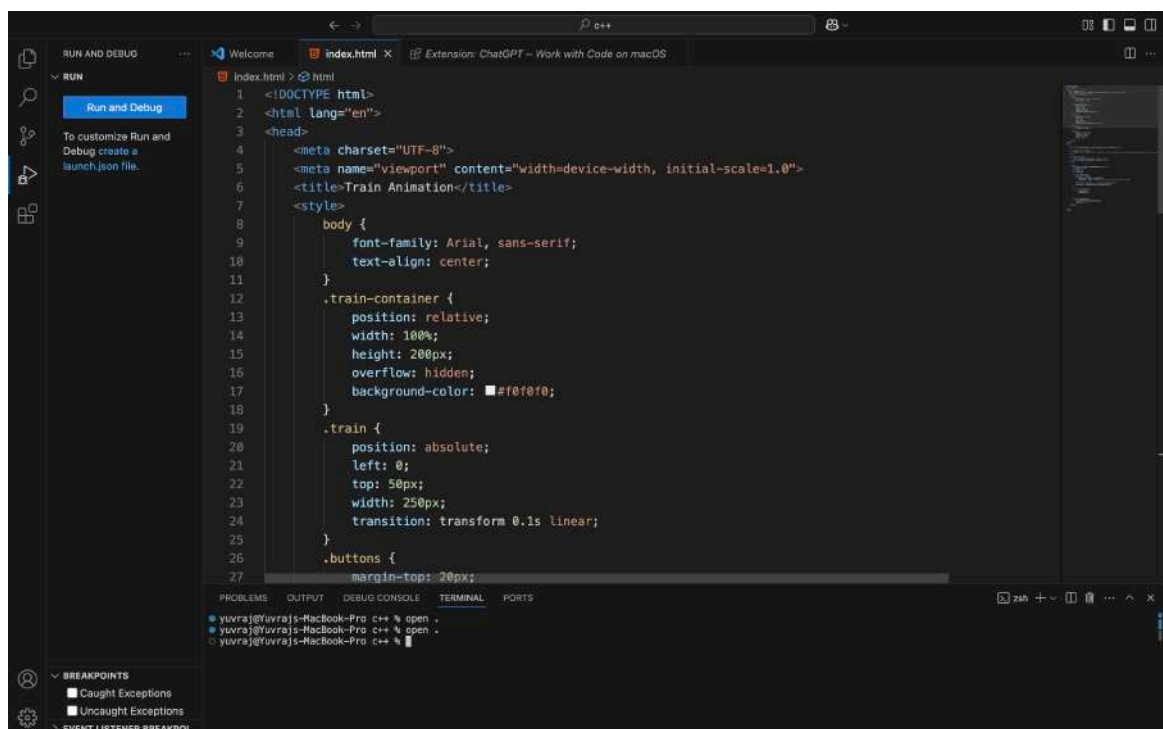


TASK - 7



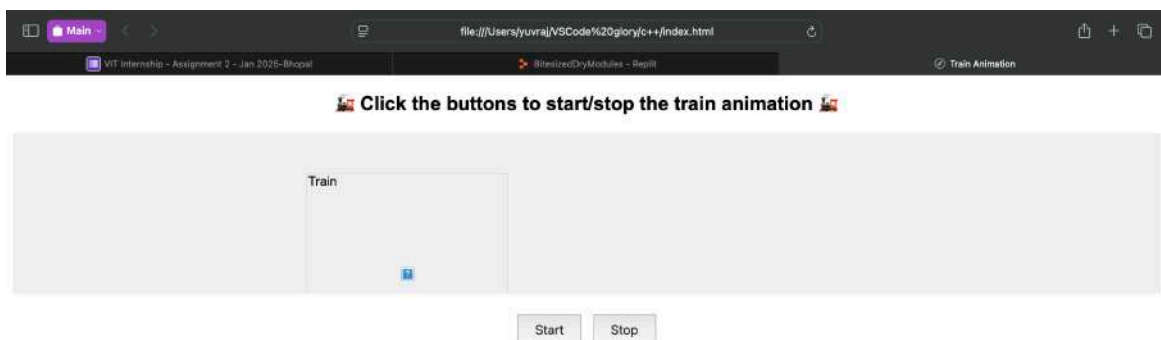
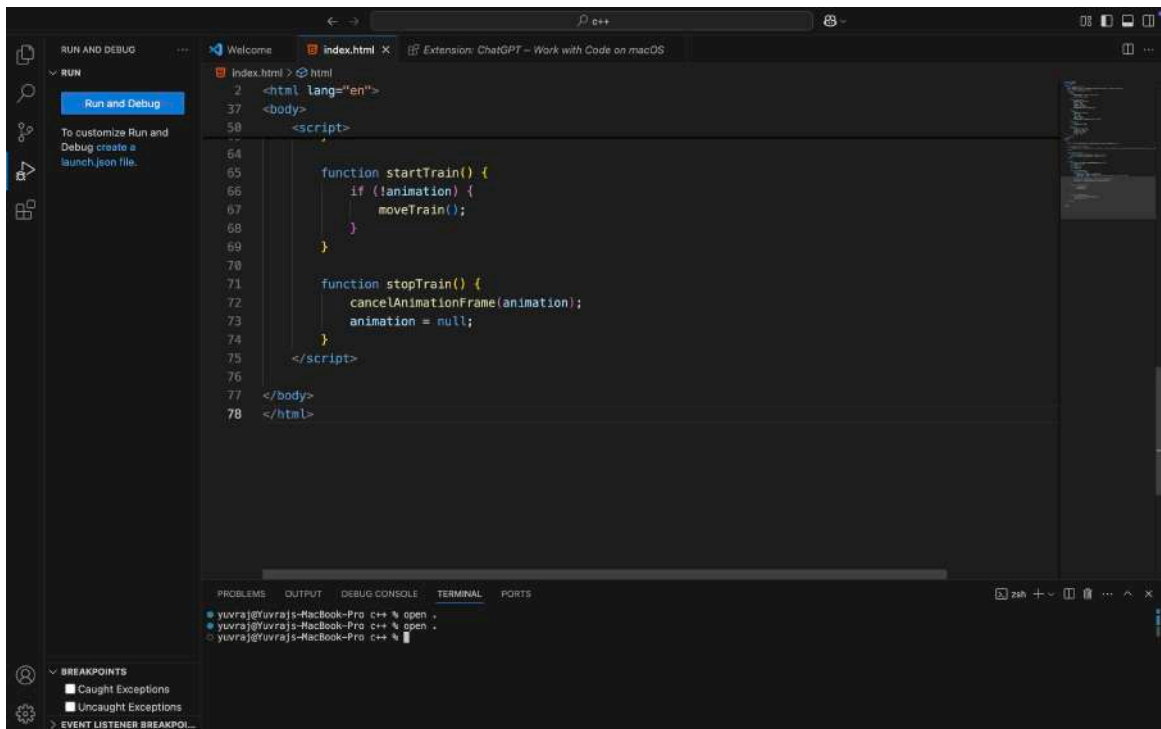
Click the following buttons to handle animation

Start Stop




```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Train Animation</title>
5 <meta charset="UTF-8">
6 </head>
7 <style>
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 .buttons {
27
28 }
29 button {
30 padding: 10px 20px;
31 font-size: 16px;
32 cursor: pointer;
33 margin: 5px;
34 }
35 </style>
36 </head>
37 <body>
38
39 <h2>Click the buttons to start/stop the train animation</h2>
40
41 <div class="train-container">
42 
43 </div>
44
45 <div class="buttons">
46 <button onclick="startTrain()">Start</button>
47 <button onclick="stopTrain()">Stop</button>
48 </div>
49 </body>
50 </html>
```

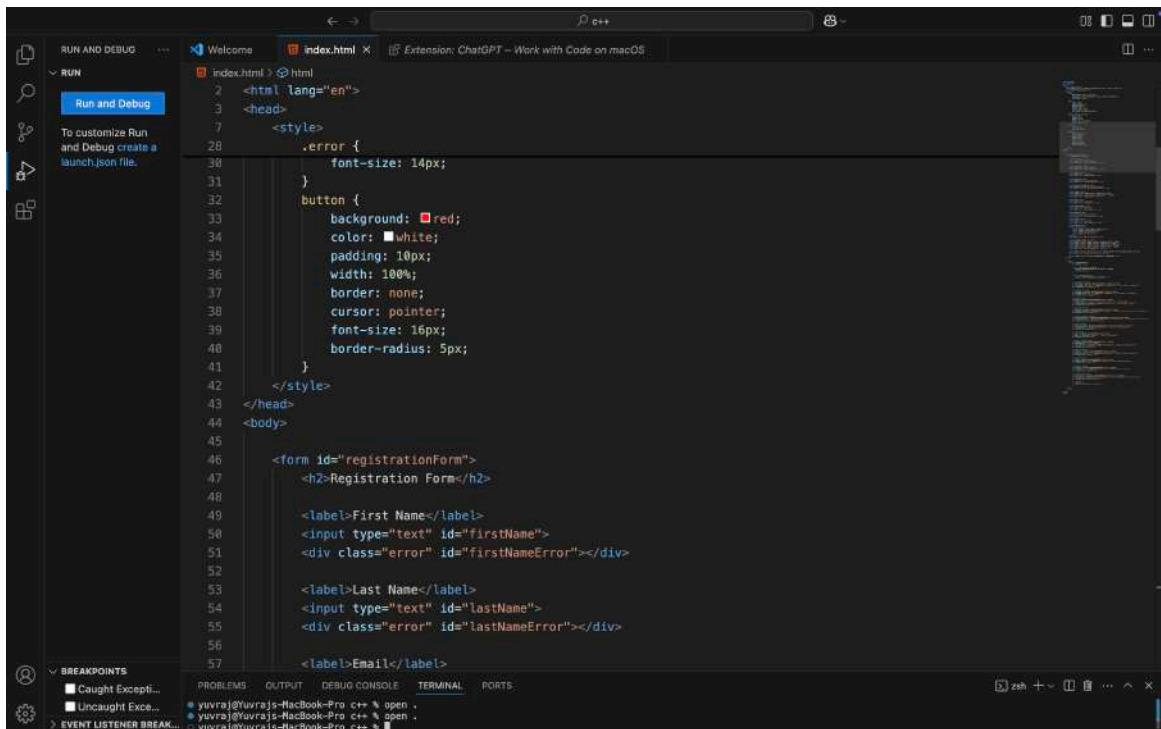
```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 <button onclick="stopTrain()">Stop</button>
48 </div>
49
50 <script>
51 let train = document.getElementById("train");
52 let position = 0;
53 let speed = 5;
54 let animation;
55
56 function moveTrain() {
57 position += speed;
58 if (position > window.innerWidth) {
59 position = -250; // Reset train position when it goes off-screen
60 }
61 train.style.transform = `translateX(${position}px)`;
62 animation = requestAnimationFrame(moveTrain);
63 }
64
65 function startTrain() {
66 if (!animation) {
67 moveTrain();
68 }
69 }
70 </script>
```

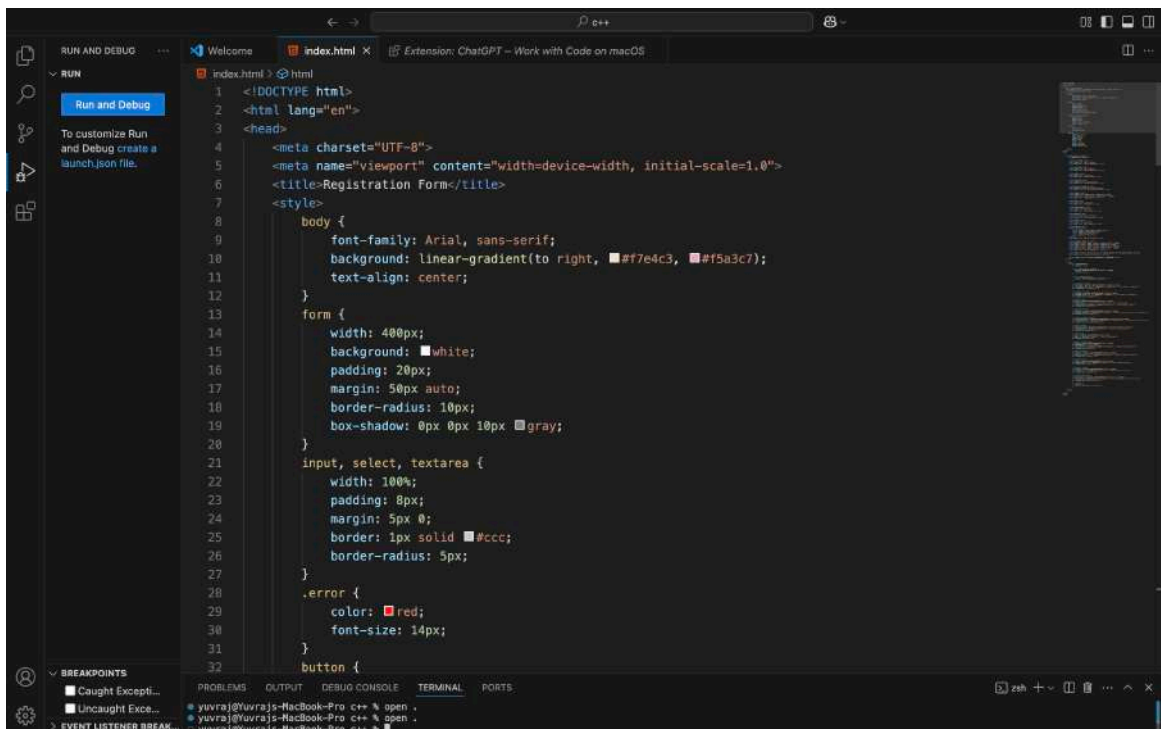
Task - 8

Registration form

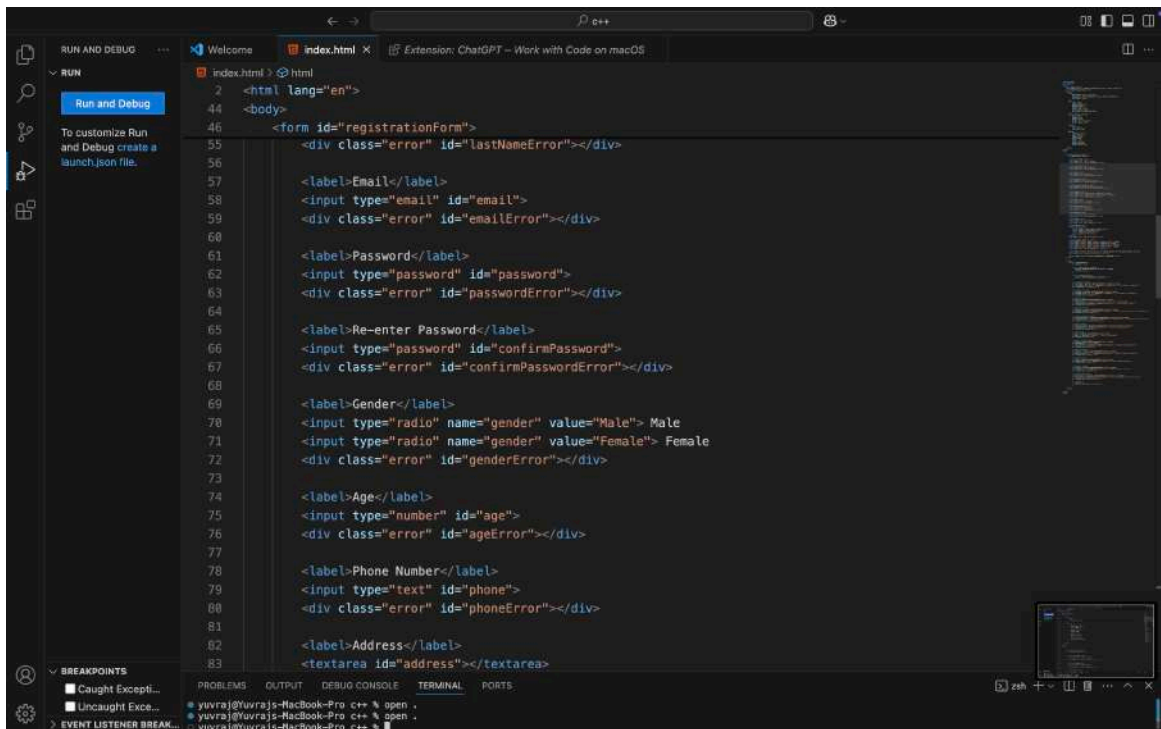
First Name	<input type="text"/>	POOR
Last Name	<input type="text"/>	POOR
Email	<input type="text"/>	POOR
Password	<input type="text"/>	POOR
Re-enter Password	<input type="text"/>	POOR OR MISMATCH
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female	
Age	<input type="text"/>	POOR
Phone Number	<input type="text"/>	POOR
Address	<input type="text"/>	POOR
State	<input type="text"/>	POOR
Country	<input type="text" value="Choose your country"/>	
Languages known	<input type="checkbox"/> English <input type="checkbox"/> Kannada <input type="checkbox"/> Hindi <input type="checkbox"/> Telugu	
<input checked="" type="checkbox"/> Hereby I declare all the given details are true		
Log in to Ethnuz website		
<input type="button" value="REGISTER"/>		



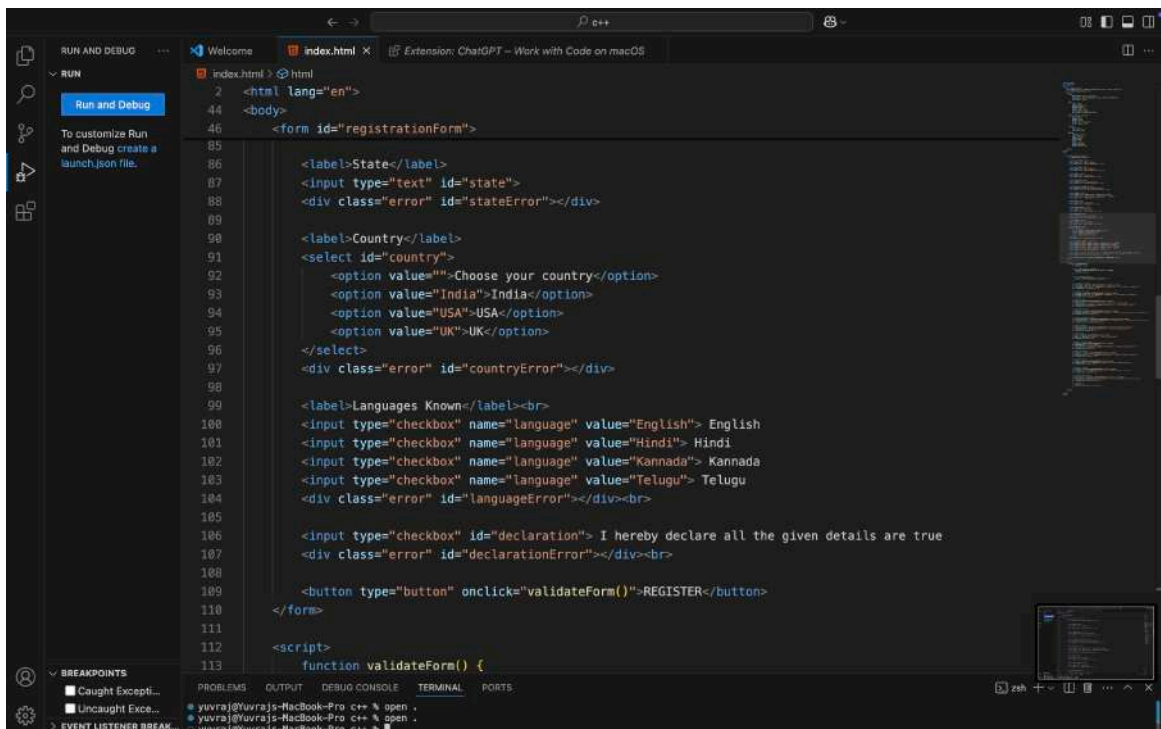
```
1 <html lang="en">
2 <head>
3
4 <style>
5
6 .error {
7     font-size: 14px;
8 }
9
10 button {
11     background: red;
12     color: white;
13     padding: 10px;
14     width: 100%;
15     border: none;
16     cursor: pointer;
17     font-size: 16px;
18     border-radius: 5px;
19 }
20 </style>
21 </head>
22 <body>
23
24 <form id="registrationForm">
25 <h2>Registration Form</h2>
26
27 <label>First Name</label>
28 <input type="text" id="firstName">
29 <div class="error" id="firstNameError"></div>
30
31 <label>Last Name</label>
32 <input type="text" id="lastName">
33 <div class="error" id="lastNameError"></div>
34
35 <label>Email</label>
```



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Registration Form</title>
7 <style>
8     body {
9         font-family: Arial, sans-serif;
10        background: linear-gradient(to right, #f7e4c3, #f5a3c7);
11        text-align: center;
12    }
13    form {
14        width: 400px;
15        background: white;
16        padding: 20px;
17        margin: 50px auto;
18        border-radius: 10px;
19        box-shadow: 0px 0px 10px gray;
20    }
21    input, select, textarea {
22        width: 100%;
23        padding: 8px;
24        margin: 5px 0;
25        border: 1px solid #ccc;
26        border-radius: 5px;
27    }
28    .error {
29        color: red;
30        font-size: 14px;
31    }
32    button {
```



```
2 <html lang="en">
44 <body>
46 <form id="registrationForm">
55 <div class="error" id="lastNameError"></div>
56
57 <label>Email</label>
58 <input type="email" id="email">
59 <div class="error" id="emailError"></div>
60
61 <label>Password</label>
62 <input type="password" id="password">
63 <div class="error" id="passwordError"></div>
64
65 <label>Re-enter Password</label>
66 <input type="password" id="confirmPassword">
67 <div class="error" id="confirmPasswordError"></div>
68
69 <label>Gender</label>
70 <input type="radio" name="gender" value="Male"> Male
71 <input type="radio" name="gender" value="Female"> Female
72 <div class="error" id="genderError"></div>
73
74 <label>Age</label>
75 <input type="number" id="age">
76 <div class="error" id="ageError"></div>
77
78 <label>Phone Number</label>
79 <input type="text" id="phone">
80 <div class="error" id="phoneError"></div>
81
82 <label>Address</label>
83 <textarea id="address"></textarea>
```



```
85
86 <label>State</label>
87 <input type="text" id="state">
88 <div class="error" id="stateError"></div>
89
90 <label>Country</label>
91 <select id="country">
92 <option value="">Choose your country</option>
93 <option value="India">India</option>
94 <option value="USA">USA</option>
95 <option value="UK">UK</option>
96 </select>
97 <div class="error" id="countryError"></div>
98
99 <label>Languages Known</label><br>
100 <input type="checkbox" name="language" value="English"> English
101 <input type="checkbox" name="language" value="Hindi"> Hindi
102 <input type="checkbox" name="language" value="Kannada"> Kannada
103 <input type="checkbox" name="language" value="Telugu"> Telugu
104 <div class="error" id="languageError"></div><br>
105
106 <input type="checkbox" id="declaration"> I hereby declare all the given details are true
107 <div class="error" id="declarationError"></div><br>
108
109 <button type="button" onclick="validateForm()">REGISTER</button>
110 </form>
111
112 <script>
113 function validateForm() {
```

The screenshot shows the Visual Studio Code editor with a file named `index.html` open. The code is a mix of HTML and JavaScript. The HTML part includes a form structure with `<html lang="en">`, `<body>`, and `</form>` tags. The JavaScript part is enclosed in a `<script>` tag and defines a `validateForm()` function. This function sets `isValid = true` and contains three sub-functions: `showError(id, message)`, `clearError(id)`, and validation logic for First Name, Last Name, and Email. The email validation uses a regular expression: `let emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;`. The interface includes a sidebar with icons for Explorer, Search, and Run and Debug, and a bottom panel with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS.

```
2 <html lang="en">
44 <body>
118 </form>
119
120 <script>
121   function validateForm() {
122     let isValid = true;
123
124     function showError(id, message) {
125       document.getElementById(id).innerText = message;
126       isValid = false;
127     }
128
129     function clearError(id) {
130       document.getElementById(id).innerText = "";
131     }
132
133     // Validate First Name
134     let firstName = document.getElementById("firstName").value;
135     if (firstName.trim() === "") showError("firstNameError", "First Name is required!");
136     else clearError("firstNameError");
137
138     // Validate Last Name
139     let lastName = document.getElementById("lastName").value;
140     if (lastName.trim() === "") showError("lastNameError", "Last Name is required!");
141     else clearError("lastNameError");
142
143     // Validate Email
144     let email = document.getElementById("email").value;
145     let emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
146     if (!emailPattern.test(email)) showError("emailError", "Invalid email format!");
147     else clearError("emailError");
148   }
149 }
150
```

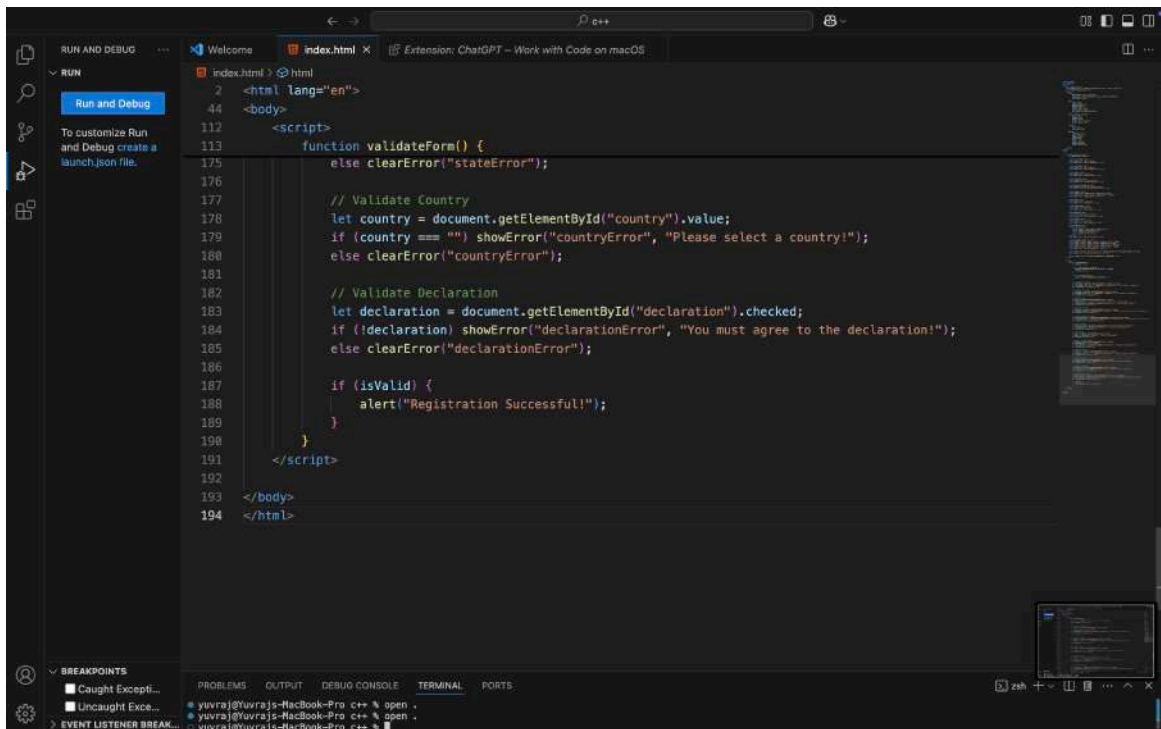
This screenshot continues the code from the previous one, showing the remaining validation logic in the `validateForm()` function. It includes validation for Password (checking length), Confirm Password (checking if it matches the password), Gender (checking if a gender is selected), Age (checking if age is greater than 0), and Phone Number. The code is as follows:

```
151 // Validate Password
152 let password = document.getElementById("password").value;
153 if (password.length < 6) showError("passwordError", "Password must be at least 6 characters!");
154 else clearError("passwordError");
155
156 // Validate Confirm Password
157 let confirmPassword = document.getElementById("confirmPassword").value;
158 if (confirmPassword !== password) showError("confirmPasswordError", "Passwords do not match!");
159 else clearError("confirmPasswordError");
160
161 // Validate Gender
162 let genderSelected = document.querySelector('input[name="gender"]:checked');
163 if (!genderSelected) showError("genderError", "Please select gender!");
164 else clearError("genderError");
165
166 // Validate Age
167 let age = document.getElementById("age").value;
168 if (age < 1) showError("ageError", "Enter a valid age!");
169 else clearError("ageError");
170
171 // Validate Phone Number
172 let phone = document.getElementById("phone").value;
173 if (!phone) showError("phoneError", "Phone number is required!");
174 else clearError("phoneError");
175 }
176
```

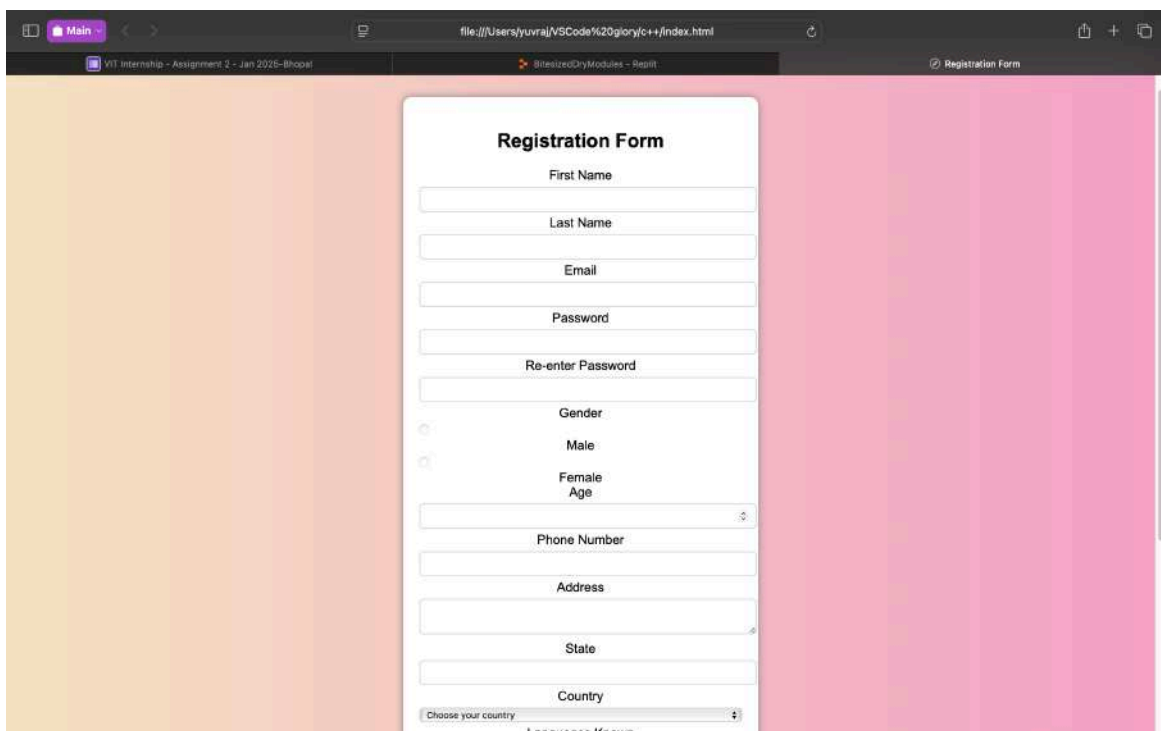


```
index.html > html
2 <html lang="en">
44 <body>
112 <script>
113     function validateForm() {
134
135         // Validate Email
136         let email = document.getElementById("email").value;
137         let emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
138         if (!emailPattern.test(email)) showError("emailError", "Invalid email format!");
139         else clearError("emailError");
140
141         // Validate Password
142         let password = document.getElementById("password").value;
143         if (password.length < 6) showError("passwordError", "Password must be at least 6 characters!");
144         else clearError("passwordError");
145
146         // Validate Confirm Password
147         let confirmPassword = document.getElementById("confirmPassword").value;
148         if (confirmPassword !== password) showError("confirmPasswordError", "Passwords do not match!");
149         else clearError("confirmPasswordError");
150
151         // Validate Gender
152         let genderSelected = document.querySelector('input[name="gender"]:checked');
153         if (!genderSelected) showError("genderError", "Please select gender!");
154         else clearError("genderError");
155
156         // Validate Age
157         let age = document.getElementById("age").value;
158         if (age < 1) showError("ageError", "Enter a valid age!");
159         else clearError("ageError");
160
161         // Validate Phone Number
```

```
index.html > html
2 <html lang="en">
44 <body>
112 <script>
113     function validateForm() {
158         if (age < 1) showError("ageError", "Enter a valid age!");
159         else clearError("ageError");
160
161         // Validate Phone Number
162         let phone = document.getElementById("phone").value;
163         let phonePattern = /^[6-9]\d{9}$/;
164         if (!phonePattern.test(phone)) showError("phoneError", "Invalid phone number!");
165         else clearError("phoneError");
166
167         // Validate Address
168         let address = document.getElementById("address").value;
169         if (address.trim() === "") showError("addressError", "Address cannot be empty!");
170         else clearError("addressError");
171
172         // Validate State
173         let state = document.getElementById("state").value;
174         if (state.trim() === "") showError("stateError", "State cannot be empty!");
175         else clearError("stateError");
176
177         // Validate Country
178         let country = document.getElementById("country").value;
179         if (country === "") showError("countryError", "Please select a country!");
180         else clearError("countryError");
181
182         // Validate Declaration
183         let declaration = document.getElementById("declaration").checked;
184         if (!declaration) showError("declarationError", "You must agree to the declaration!");
185         else clearError("declarationError");
```



```
index.html > html
2 <html lang="en">
44 <body>
112 <script>
113   function validateForm() {
175     else clearError("stateError");
176
177     // Validate Country
178     let country = document.getElementById("country").value;
179     if (country == "") showError("countryError", "Please select a country!");
180     else clearError("countryError");
181
182     // Validate Declaration
183     let declaration = document.getElementById("declaration").checked;
184     if (!declaration) showError("declarationError", "You must agree to the declaration!");
185     else clearError("declarationError");
186
187     if (isValid) {
188       alert("Registration Successful!");
189     }
190   }
191 </script>
192
193 </body>
194 </html>
```



Registration Form

First Name

Last Name

Email

Password

Re-enter Password

Gender

Male

Female

Age

Phone Number

Address

State

Country

Choose your country

file:///Users/yuvraj/VSCode%20glory/c++/index.html

Registration Form

Password

Re-enter Password

Gender

☐ Male

☐ Female

Age

Phone Number

Address

State

Country

Choose your country

Languages Known

☐ English

☐ Hindi

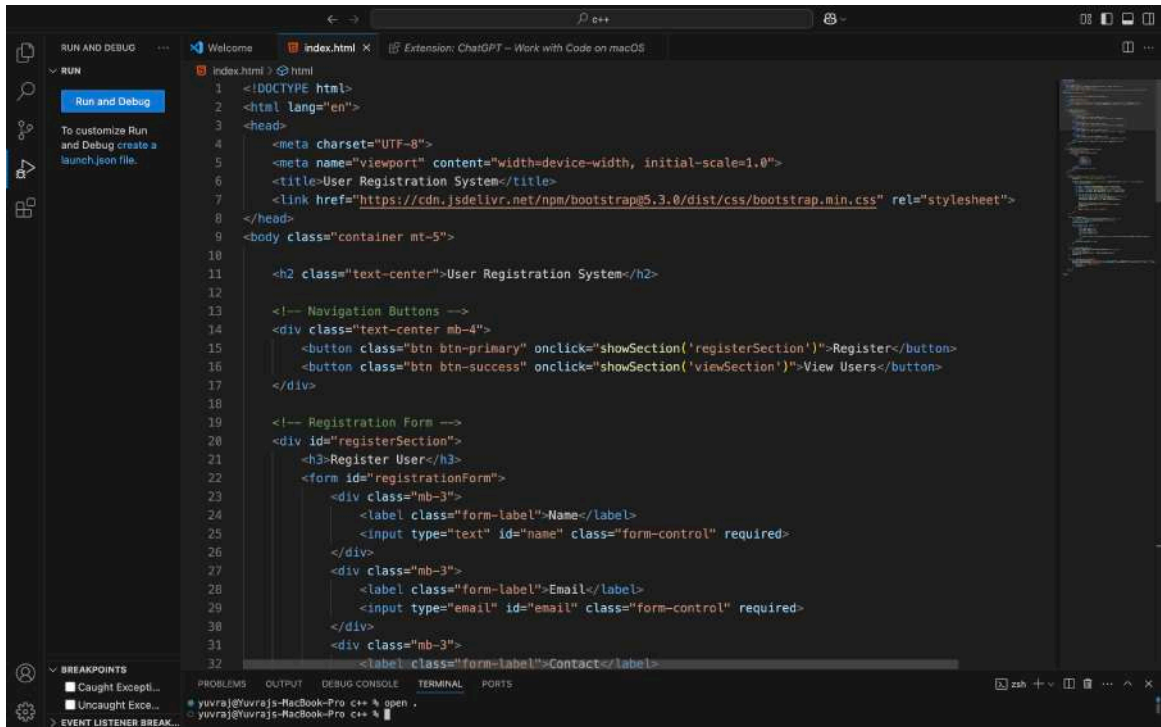
☐ Kannada

☐ Telugu

☐ I hereby declare all the given details are true

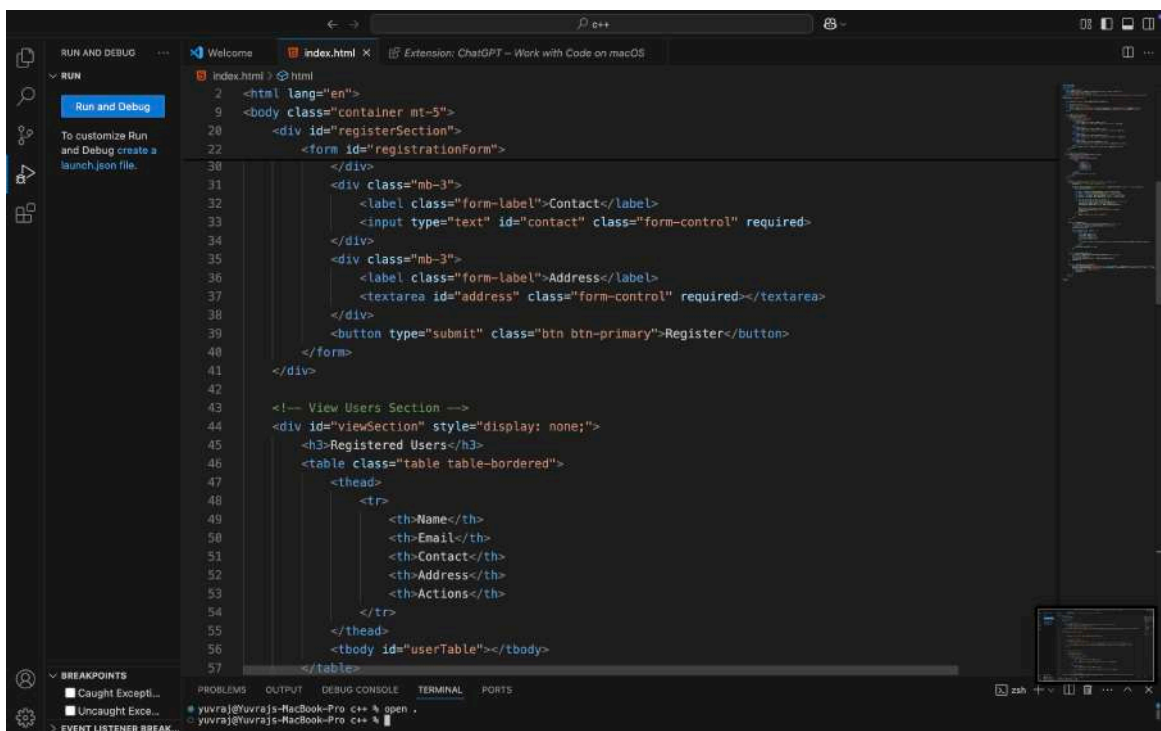
REGISTER

TAKS - 9



The screenshot shows the VS Code editor with a file named `index.html`. The code is written in HTML and includes Bootstrap 5.3.0 CSS. It features a navigation bar with 'Register' and 'View Users' buttons. Below the navigation bar is a registration form with fields for Name, Email, and Contact, each with a 'required' attribute. The form is enclosed in a `<div id='registerSection'>` container.

```
1 <!DOCTYPE html>
2 <html lang='en'>
3 <head>
4   <meta charset='UTF-8'>
5   <meta name='viewport' content='width=device-width, initial-scale=1.0'>
6   <title>User Registration System</title>
7   <link href='https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css' rel='stylesheet'>
8 </head>
9 <body class='container mt-5'>
10
11   <h2 class='text-center'>User Registration System</h2>
12
13   <!-- Navigation Buttons -->
14   <div class='text-center mb-4'>
15     <button class='btn btn-primary' onclick='showSection('registerSection')'>Register</button>
16     <button class='btn btn-success' onclick='showSection('viewSection')'>View Users</button>
17   </div>
18
19   <!-- Registration Form -->
20   <div id='registerSection'>
21     <h3>Register User</h3>
22     <form id='registrationForm'>
23       <div class='mb-3'>
24         <label class='form-label'>Name</label>
25         <input type='text' id='name' class='form-control' required>
26       </div>
27       <div class='mb-3'>
28         <label class='form-label'>Email</label>
29         <input type='email' id='email' class='form-control' required>
30       </div>
31       <div class='mb-3'>
32         <label class='form-label'>Contact</label>
```

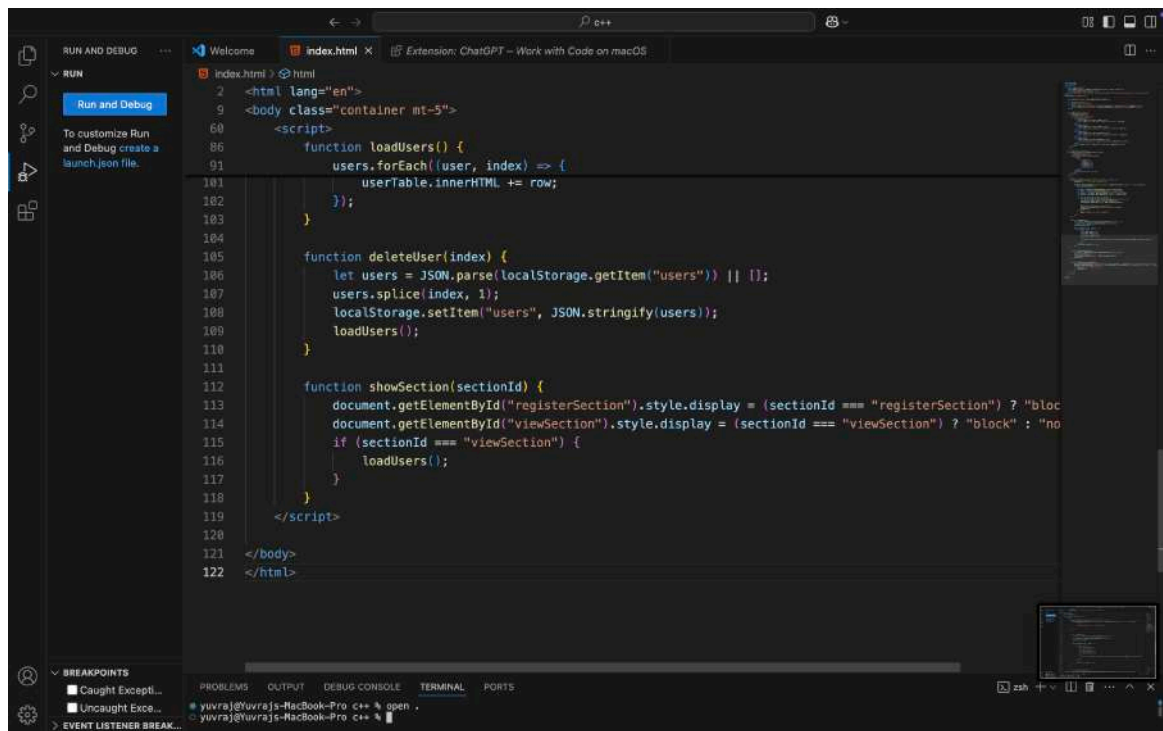


The screenshot shows the continuation of the `index.html` file. It includes the 'View Users' section, which is currently hidden (`style='display: none;'`). This section contains a table titled 'Registered Users' with columns for Name, Email, Contact, Address, and Actions. The table has a header row and a body row with placeholder data. The form from the previous section is also visible, showing the 'Contact' field and the 'Register' button.

```
33       <input type='text' id='contact' class='form-control' required>
34     </div>
35     <div class='mb-3'>
36       <label class='form-label'>Address</label>
37       <textarea id='address' class='form-control' required></textarea>
38     </div>
39     <button type='submit' class='btn btn-primary'>Register</button>
40   </form>
41 </div>
42
43   <!-- View Users Section -->
44   <div id='viewSection' style='display: none;'>
45     <h3>Registered Users</h3>
46     <table class='table table-bordered'>
47       <thead>
48         <tr>
49           <th>Name</th>
50           <th>Email</th>
51           <th>Contact</th>
52           <th>Address</th>
53           <th>Actions</th>
54         </tr>
55       </thead>
56       <tbody id='userTable'></tbody>
57     </table>
```

```
2 <html lang="en">
9 <body class="container mt-5">
44 <div id="viewSection" style="display: none;">
57 </div>
58 </div>
59
60 <script>
61   document.addEventListener("DOMContentLoaded", function () {
62     loadUsers(); // Load users on page load
63
64     document.getElementById("registrationForm").addEventListener("submit", function (event) {
65       event.preventDefault();
66
67       let name = document.getElementById("name").value.trim();
68       let email = document.getElementById("email").value.trim();
69       let contact = document.getElementById("contact").value.trim();
70       let address = document.getElementById("address").value.trim();
71
72       if (name && email && contact && address) {
73         let users = JSON.parse(localStorage.getItem("users")) || [];
74         users.push({ name, email, contact, address });
75         localStorage.setItem("users", JSON.stringify(users));
76
77         alert("User registered successfully!");
78         document.getElementById("registrationForm").reset();
79         loadUsers();
80       } else {
81         alert("Please fill in all fields!");
82       }
83     });
84   });
85
```

```
60 <script>
61   document.addEventListener("DOMContentLoaded", function () {
62     document.getElementById("registrationForm").addEventListener("submit", function (event) {
63       event.preventDefault();
64       alert("Please fill in all fields!");
65     });
66   });
67
68   function loadUsers() {
69     let users = JSON.parse(localStorage.getItem("users")) || [];
70     let userTable = document.getElementById("userTable");
71     userTable.innerHTML = "";
72
73     users.forEach((user, index) => {
74       let row = `<tr>
75         <td>${user.name}</td>
76         <td>${user.email}</td>
77         <td>${user.contact}</td>
78         <td>${user.address}</td>
79         <td>
80           <button class="btn btn-danger btn-sm" onclick="deleteUser(${index})">Delete</button>
81         </td>
82       </tr>`;
83       userTable.innerHTML += row;
84     });
85
86     function deleteUser(index) {
87       let users = JSON.parse(localStorage.getItem("users")) || [];
88     }
89   }
90
```



User Registration System

RegisterView Users

Register User

Name

Email

Contact

Address

Register

Registered Users

Name	Email	Contact	Address
Shankar	shankar@gmail.com	98884534332	Bangalore, Karnataka

[Register More Users](#)