Experiment No. 05

Title: Implement Javascript validation for Website Forms.

Batch:A2 **Roll No.:1600421063** **Experiment No.:5**

Aim: To Implement Javascript validation for Website Forms.
Resources needed: Notepad++, Web Browser
Theory:

JavaScript is a scripting language produced by Netscape for use within HTML Web pages. JavaScript is loosely based on Java and it is built into all the major modern browsers. JavaScript is a lightweight, interpreted programming language, Complementary to and integrated with Java , Complementary to and integrated with HTML , Open and cross- platform and is case sensitive.

Data validation is the process of ensuring that user input is clean, correct, and useful. Typical validation tasks are:

- Has the user filled in all required fields?
- Has the user entered a valid date?
- Has the user entered text in a numeric field?
- Most often, the purpose of data validation is to ensure correct user input.

Validation can be defined by many different methods, and deployed in many different ways.

- Server side validation is performed by a web server, after input has been sent to the server.
- Client side validation is performed by a web browser, before input is sent to a web server.

For example HTML form validation can be done by JavaScript. If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

```
function validateForm() {

 var x =

 document.forms["myForm"]["fname"].value; if (x

 == "") {

  alert("Name must be filled out");

  return false;
```

```
        }
      }
```

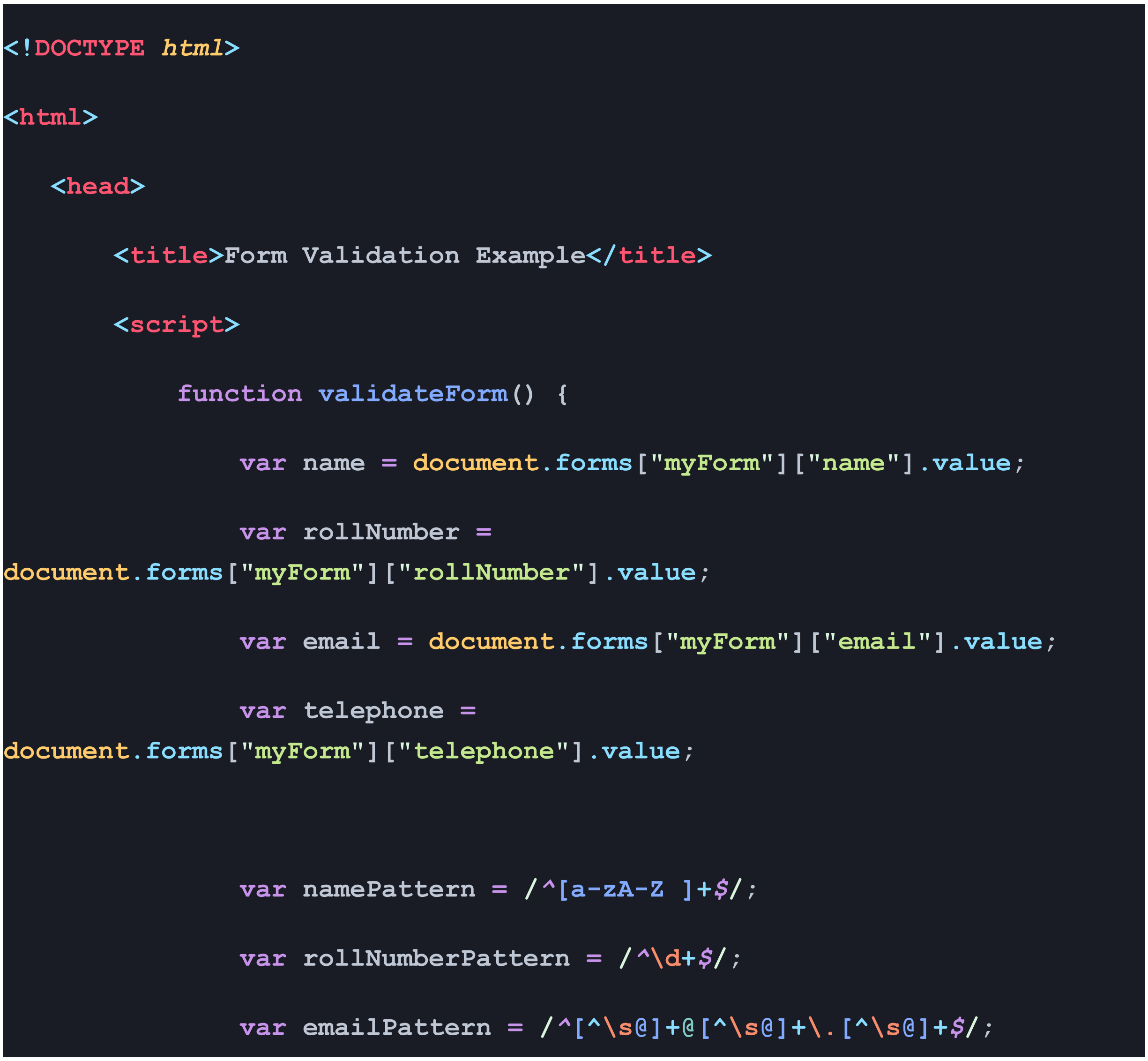
Activity:

Add validations for the Website Forms Such as

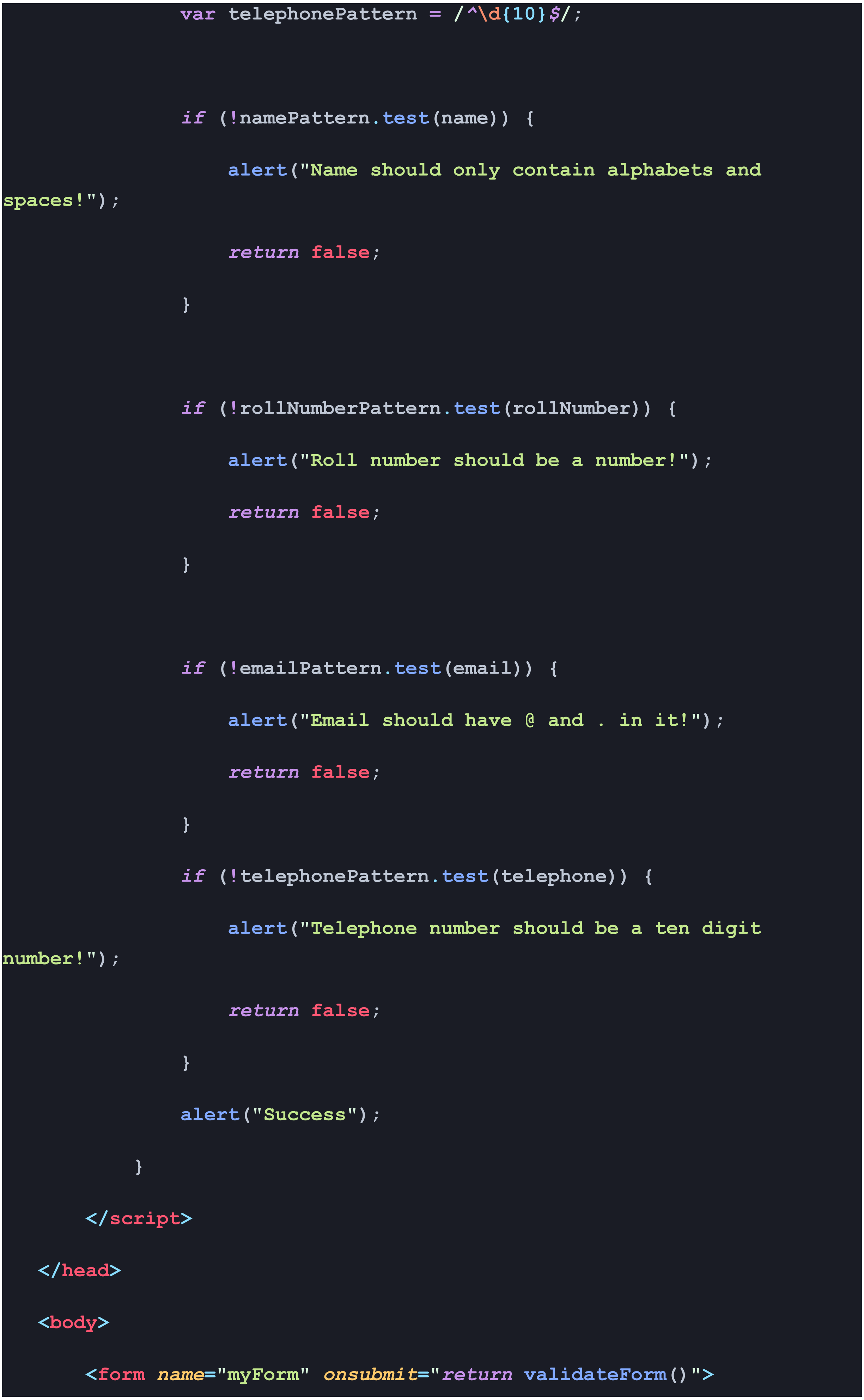
1) Name should string
2) Roll number should a number
3) Email id should have @ and . in it
4) Telephone number should be a ten digit number.

Students need to add various validations to their form input as per the requirement of the user interface.

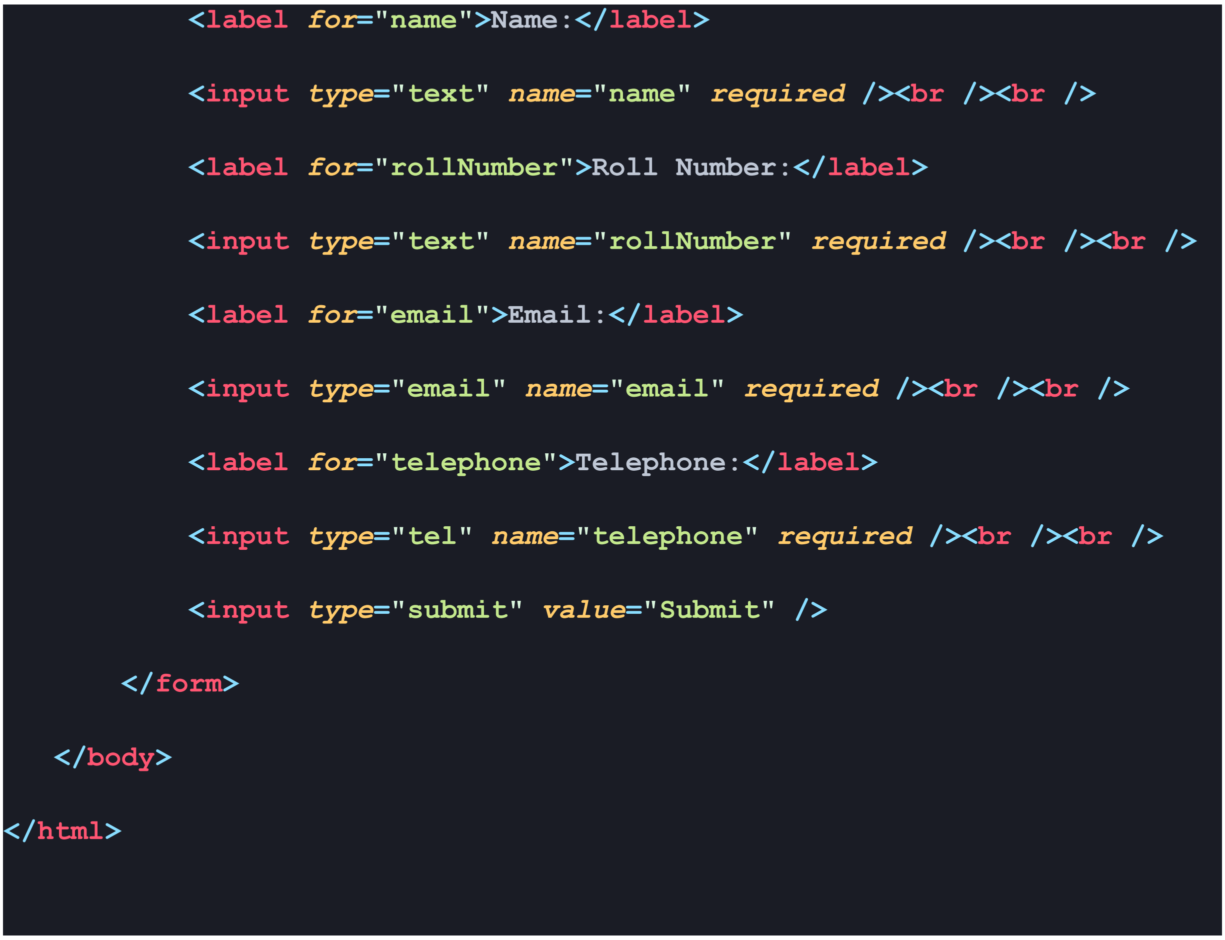
Results: (Program printout with output)



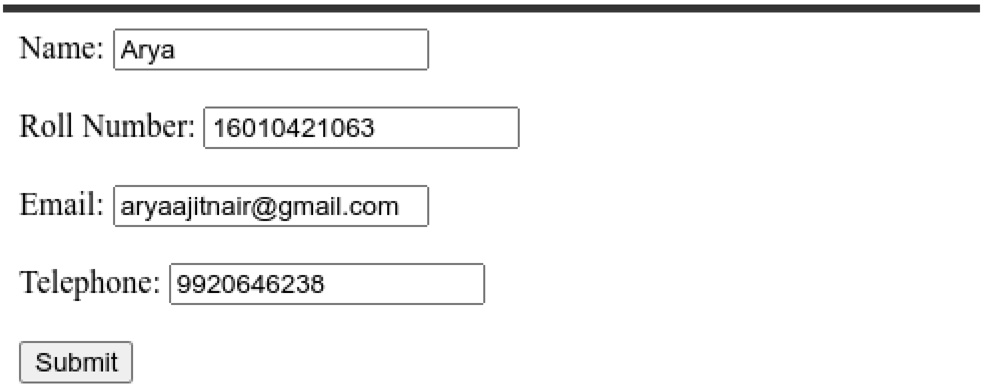
```
<!DOCTYPE html>
<html>
   <head>
       <title>Form Validation Example</title>
       <script>
           function validateForm() {
               var name = document.forms["myForm"]["name"].value;
               var rollNumber =
document.forms["myForm"]["rollNumber"].value;
               var email = document.forms["myForm"]["email"].value;
               var telephone =
document.forms["myForm"]["telephone"].value;

               var namePattern = /^[a-zA-Z ]+$/;
               var rollNumberPattern = /^\d+$/;
               var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```



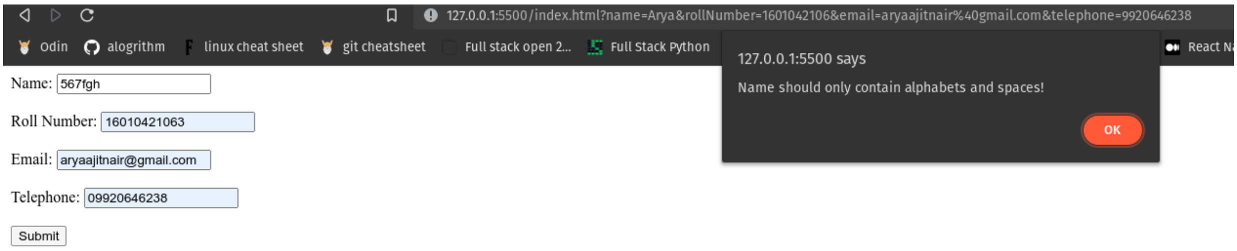
```
                var telephonePattern = /^\d{10}$/;


                if (!namePattern.test(name)) {
                    alert("Name should only contain alphabets and
spaces!");
                    return false;
                }


                if (!rollNumberPattern.test(rollNumber)) {
                    alert("Roll number should be a number!");
                    return false;
                }


                if (!emailPattern.test(email)) {
                    alert("Email should have @ and . in it!");
                    return false;
                }
                if (!telephonePattern.test(telephone)) {
                    alert("Telephone number should be a ten digit
number!");
                    return false;
                }
                alert("Success");
            }
        </script>
    </head>
    <body>
        <form name="myForm" onsubmit="return validateForm()">
```



```
            <label for="name">Name:</label>
            <input type="text" name="name" required /><br /><br />
            <label for="rollNumber">Roll Number:</label>
            <input type="text" name="rollNumber" required /><br /><br />
            <label for="email">Email:</label>
            <input type="email" name="email" required /><br /><br />
            <label for="telephone">Telephone:</label>
            <input type="tel" name="telephone" required /><br /><br />
            <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```



Name:	Arya
Roll Number:	16010421063
Email:	aryaajitnair@gmail.com
Telephone:	9920646238

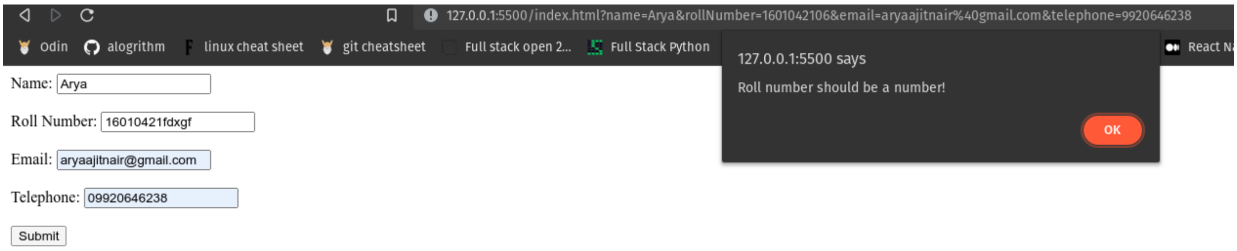


127.0.0.1:5500/index.html?name=Arya&rollNumber=16010421063&email=aryaajitnair%40gmail.com&telephone=9920646238
Odin
alogrithm
linux cheat sheet
git cheatsheet
Full stack open 2...
Full Stack Python
React N

Name:	567fgh
Roll Number:	16010421063
Email:	aryaajitnair@gmail.com
Telephone:	09920646238

127.0.0.1:5500 says

Name should only contain alphabets and spaces!



127.0.0.1:5500/index.html?name=Arya&rollNumber=1601042106&email=aryaajitnair%40gmail.com&telephone=9920646238

127.0.0.1:5500 says

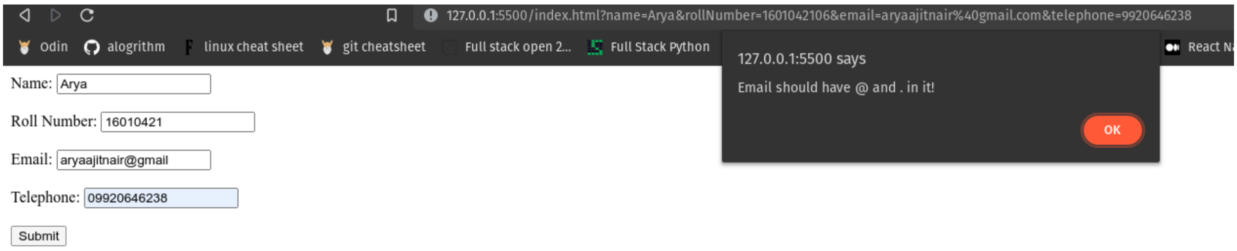
Roll number should be a number!

Name: Arya

Roll Number: 16010421fdxgf

Email: aryaajitnair@gmail.com

Telephone: 09920646238



127.0.0.1:5500/index.html?name=Arya&rollNumber=1601042106&email=aryaajitnair%40gmail.com&telephone=9920646238

127.0.0.1:5500 says

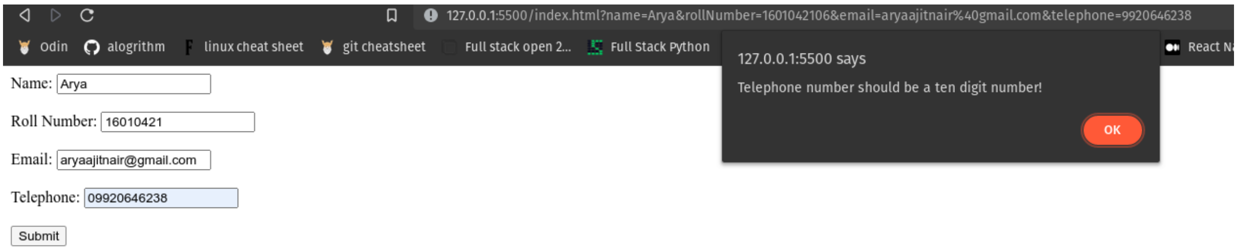
Email should have @ and . in it!

Name: Arya

Roll Number: 16010421

Email: aryaajitnair@gmail

Telephone: 09920646238



127.0.0.1:5500/index.html?name=Arya&rollNumber=1601042106&email=aryaajitnair%40gmail.com&telephone=9920646238

127.0.0.1:5500 says

Telephone number should be a ten digit number!

Name: Arya

Roll Number: 16010421

Email: aryaajitnair@gmail.com

Telephone: 09920646238

Questions:

Q1) **Why to carry out Validation at client side using scripting language?**

Client-side validation using scripting languages like JavaScript is important because it provides immediate feedback to the user, reduces server load, and improves the user experience. By validating user input on the client-side, errors can be identified and corrected quickly without waiting for the server to respond. This helps to prevent unnecessary server requests and network traffic, enhancing the security of the application. Additionally, client-side validation can improve the user experience by reducing the frustration of submitting incorrect or incomplete data, resulting in a more efficient and user-friendly application.

Q2) What is the difference between client side validation and server side validation?

Client-side validation and server-side validation differ in their location and timing of execution. Client-side validation happens on the user's device, typically within a web browser, while server-side validation happens on the server, after the form data is submitted. Client-side validation provides immediate feedback to the user, improving the user experience and reducing server load. However, client-side validation can be bypassed by users who disable JavaScript or manipulate the form data before submission. Server-side validation, on the other hand, is more secure as it ensures that the data is validated and sanitized before being processed by the server. However, it may take longer for feedback to be provided to the user, and it may require additional server resources to handle the validation. A good approach is to use a combination of both client-side and server-side validation for the best user experience and security.

Outcomes:CO 3 Apply JavaScript and JSON for Web Application development

Conclusion: (Conclusion to be based on the outcomes achieved)
Successfully implemented validation using javascript

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- "Web technologies: Black Book", Dreamtech Publications
- http://www.w3schools.com