**Web X.0 Lab**

**Experiment - 6**

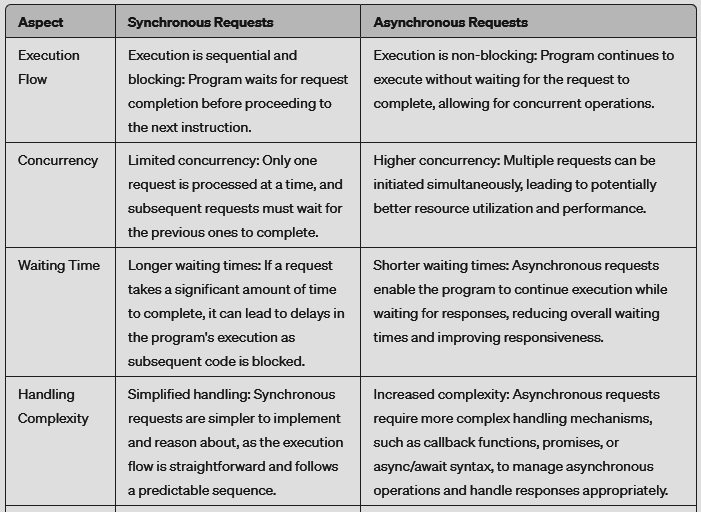
Name: Shashwat Tripathi

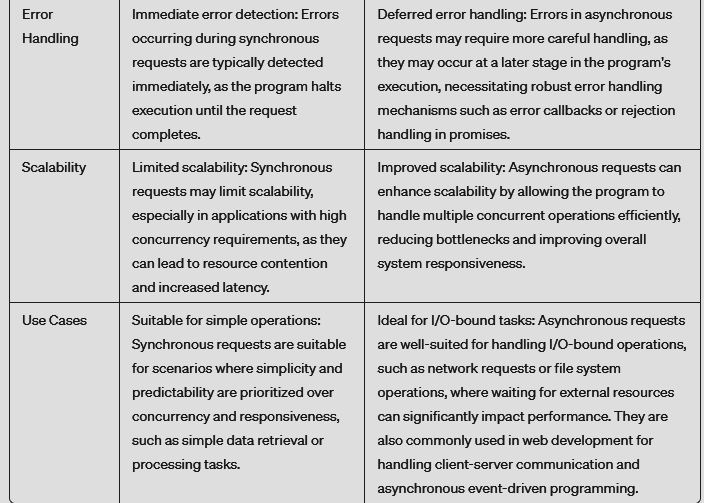
Div: D15A RollNo: 64

**Aim: To study AJAX**

**Theory:**

1. **How do Synchronous and Asynchronous Requests differ?**





1. **Describe various properties and methods used in XMLHttpRequest Object**

Various properties and methods used in the XMLHttpRequest object are:-

1. The onreadystatechange property is an event handler triggered whenever the readyState property changes.
2. readyState indicates the current state of the XMLHttpRequest object, ranging from 0 to 4.
3. status holds the HTTP status code returned by the server.
4. statusText contains the textual description of the HTTP status code.
5. response holds the response body retrieved from the server.
6. responseText contains the response body as a string.
7. responseType specifies the expected type of response from the server.
8. responseURL contains the URL of the response returned by the server.
9. timeout specifies the maximum time for a request before it times out.
10. open(method, url, async) initializes a request with HTTP method, URL, and async flag.
11. send(data) sends the request to the server.
12. setRequestHeader(name, value) sets the value of an HTTP request header.
13. abort() cancels the current request.
14. getAllResponseHeaders() returns all response headers.
15. getResponseHeader(name) returns the value of a specific response header.
16. overrideMimeType(mimeType) overrides the MIME type returned by the server.
17. addEventListener(event, callback) adds an event listener.
18. removeEventListener(event, callback) removes an event listener.

**Problem Statement:**

**Create a registration page having fields like Name, College, Username and Password (read password twice).**

**Validate the form by checking for:**

**Code:**

**/server.js**

import users from './db/users';

import colleges from './db/colleges';

app.post('/register', (req, res) => {

const { name, college, username, password } = req.body;

if (!name.trim()) {

res.status(400).send('Name field is required.');

return;

}

if (!college.trim()) {

res.status(400).send('College field is required.');

return;

}

if (password !== req.body.confirmPassword) {

res.status(400).send('Passwords do not match.');

return;

}

// Check if username already exists

if (users.some(user => user.username === username)) {

res.status(400).send('Username is already taken.');

return;

}

res.send('Registration successful!');

});

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registration Form</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h2>Registration Form</h2>

<form id="registrationForm">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="college">College:</label>

<input type="text" id="college" name="college" required list="collegeList">

<datalist id="collegeList" name="collegeList" >

<select id="collegeDropdown"></select>

</datalist>

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword" required>

<button type="submit">Register</button>

</form>

<p id="message"></p>

</div>

<script src="script.js"></script>

</body>

</html>

**script.js**

function validateForm(event) {

event.preventDefault();

const name = document.getElementById('name').value;

const college = document.getElementById('college').value;

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

const confirmPassword = document.getElementById('confirmPassword').value;

if (name.trim() === '') {

showMessage('Name field is required.');

return;

}

if (password !== confirmPassword) {

showMessage('Passwords do not match.');

return;

}

const formData = new FormData();

formData.append('name', name);

formData.append('college', college);

formData.append('username', username);

formData.append('password', password);

// Send AJAX request to the server

const xhr = new XMLHttpRequest();

xhr.open('POST', 'register.php'); // Specify the server-side script for form submission

xhr.onreadystatechange = function () {

if (xhr.readyState === 4) {

if (xhr.status === 200) {

showMessage(xhr.responseText);

} else {

showMessage('An error occurred while processing your request.');

}

}

};

xhr.send(formData);

}

function showMessage(message) {

const messageElement = document.getElementById('message');

messageElement.textContent = message;

}

function suggestColleges(input) {

const suggestions = colleges.filter(college =>

college.toLowerCase().startsWith(input.toLowerCase())

);

const collegeDropdown = document.getElementById('collegeDropdown');

collegeDropdown.innerHTML = '';

suggestions.forEach(college => {

const option = document.createElement('option');

option.value = college;

collegeDropdown.appendChild(option);

});

}

document.getElementById('college').addEventListener('input', function() {

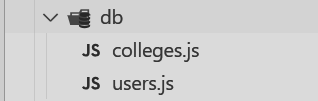
const input = this.value;

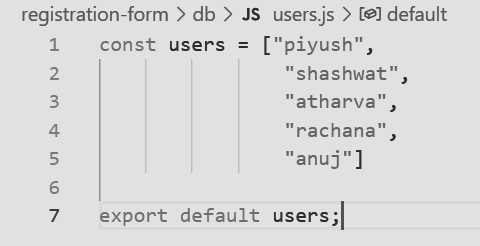
suggestColleges(input);

});

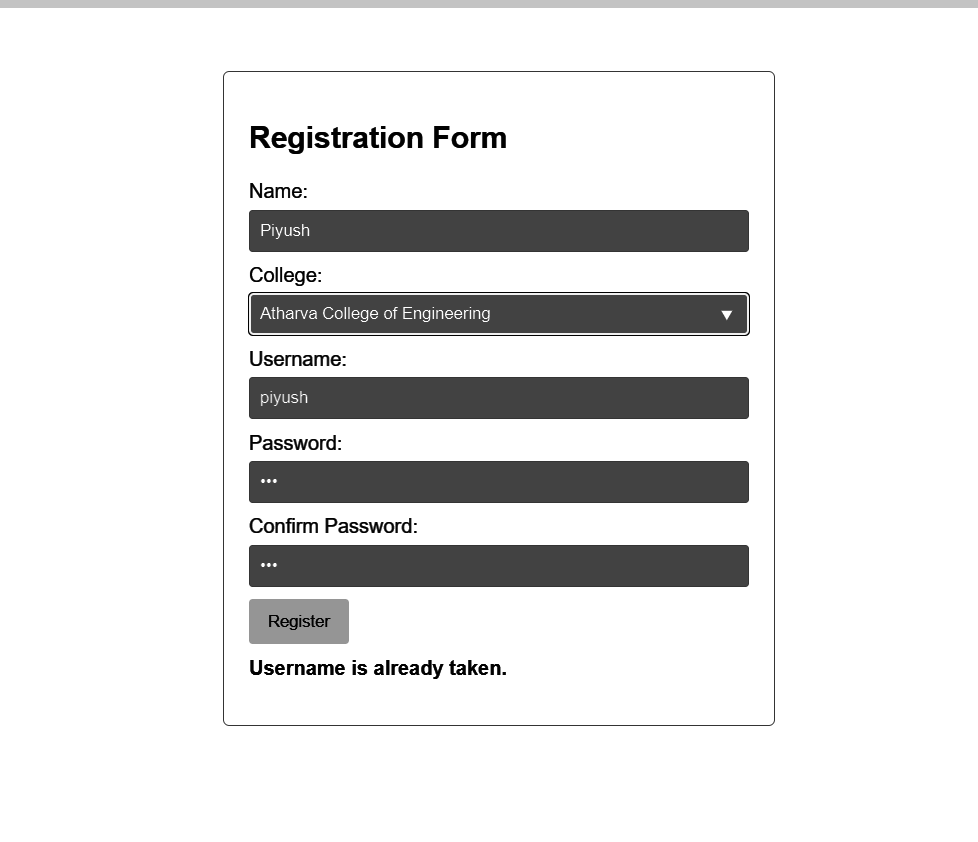
document.getElementById('registrationForm').addEventListener('submit', validateForm);

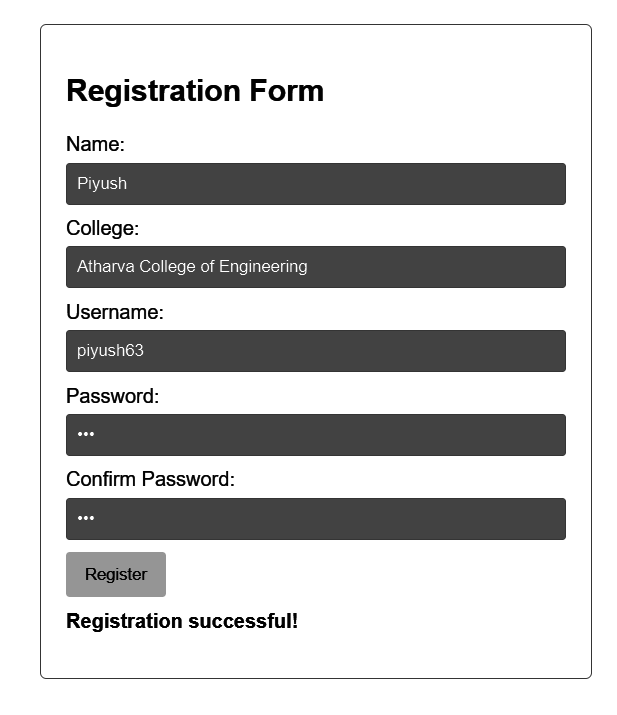
1. Username is not same as existing entries



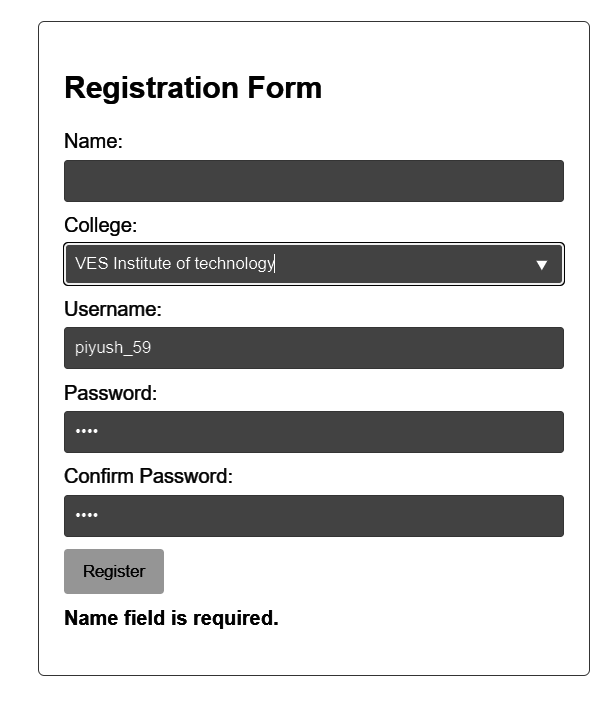


List of usernames



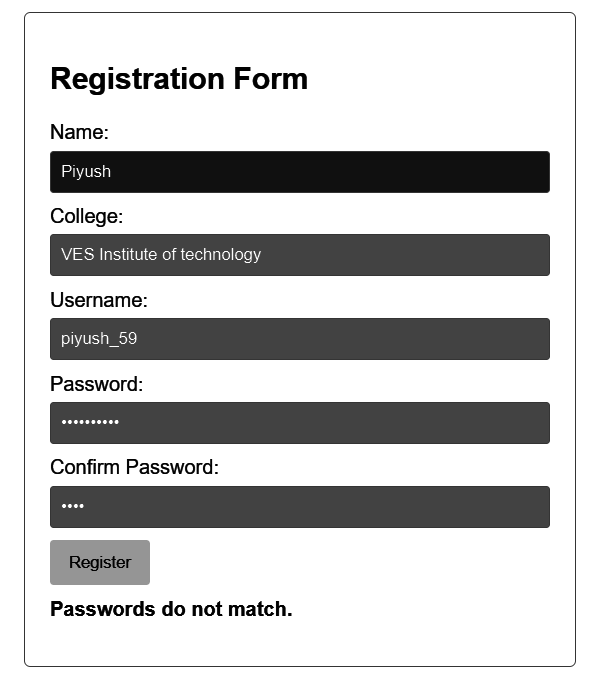


2. Name field is not empty

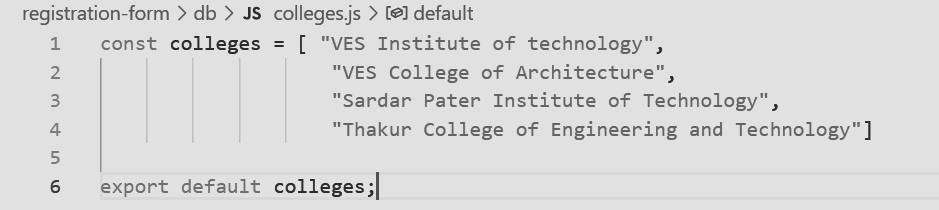


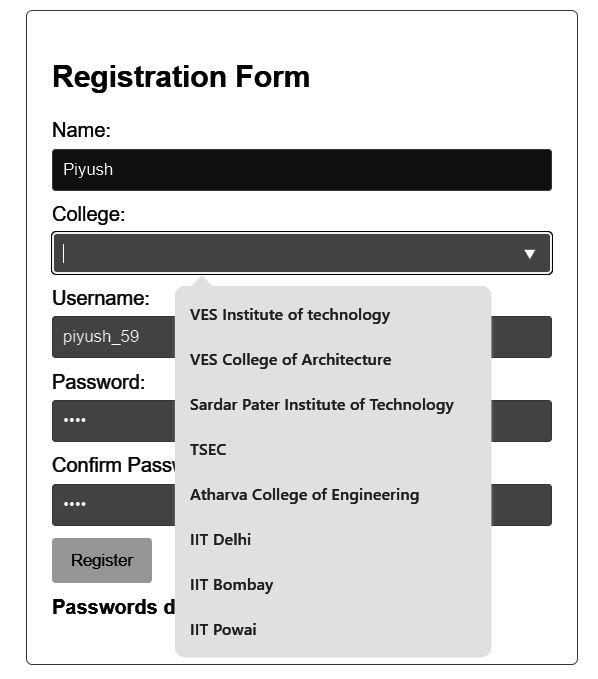
3. Retyped password is matching with the earlier one. Prompt a message is

And also auto suggest college names.

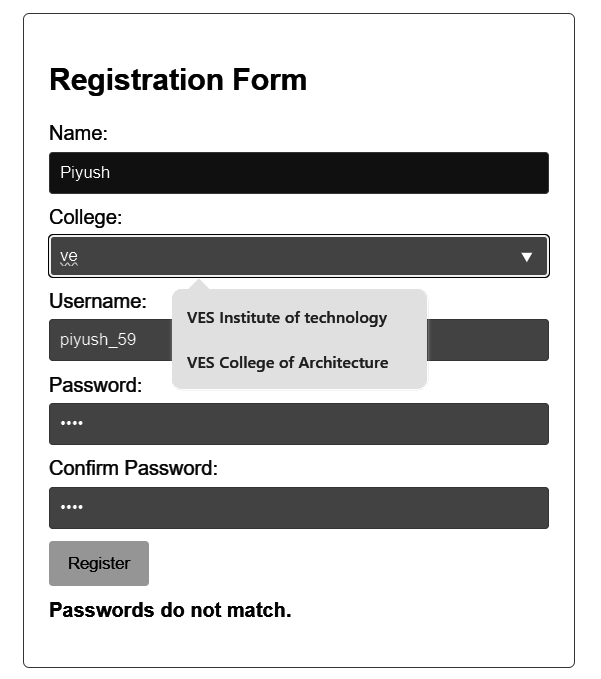


Password mismatch



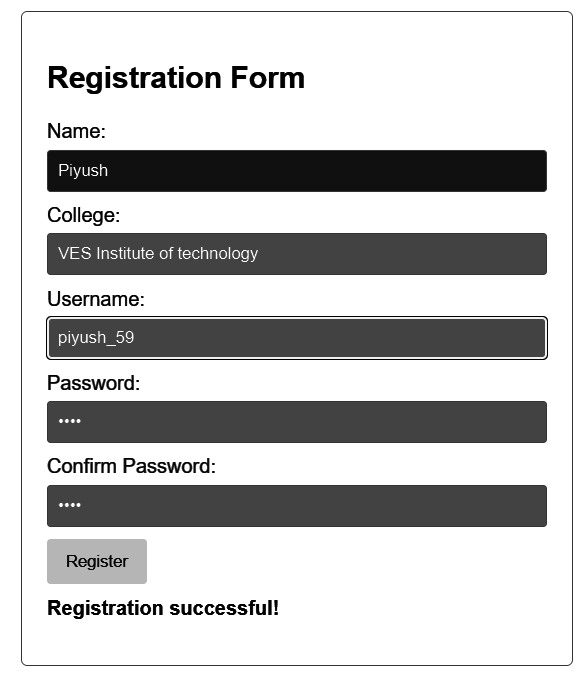


Before searching all the colleges are displayed



Upon searching, only matching colleges are suggested

Show the message "Successfully Registered" on the same page below the submit button, on Successfully registration. Let all the updations on the page be Asynchronously loaded. Implement the same using XMLHttpRequest Object.



Successful registration

**Conclusion:** Thus, we studied the concepts of Asynchronous Javascript XML(AJAX) and implemented the same to solve the above given problem statement.