EXPERIMENT NO. 9 - AJAX

Name of Student	**Ronak Katariya**
Class Roll No	**D15A/23**
D.O.P.	**20/03/2025**
D.O.S.	**27/03/2025**
Sign and Grade	

1) **Aim:** To study AJAX
2) **Theory:**

A. How do Synchronous and Asynchronous Requests differ?

Synchronous Request:

A synchronous request blocks the execution of code until the response is received from the server.

During this time, the browser becomes unresponsive (i.e., the user interface freezes) because it waits for the server response.

It is generally not recommended in web applications due to its poor user experience.

Asynchronous Request:

An asynchronous request allows the execution of code to continue while the request is being processed in the background.

The user can continue interacting with the page, making the web application faster and more responsive.

AJAX (Asynchronous JavaScript and XML) uses asynchronous requests to fetch or send data without reloading the page.

B. Describe various properties and methods used in XMLHttpRequest Object

Property	Description
`readyState`	Holds the status of the request (0 to 4)
`status`	Returns the HTTP status code (e.g., 200, 404)
`statusText`	Status in text form (e.g., "OK", "Not Found")
`responseText`	Returns the response as a string
`responseXML`	Returns the response as XML
`onreadystatechange`	Event triggered when `readyState` changes

Method	Description
`open(method, url, async)`	Initializes the request. `method` can be GET or POST. `async` is a boolean (true = asynchronous).
`send(data)`	Sends the request to the server. For GET, pass `null`; for POST, pass the data.
`setRequestHeader(header, value)`	Sets custom headers before sending the request (e.g., `Content-Type`)
`abort()`	Cancels the current request

3) **Problem Statement:**

Create a registration page having fields like Name, College, Username and Password (read password twice).
Validate the form by checking for
1. Usernameis not same as existing entries
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.
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.

4) **Output:**

```
<!DOCTYPE html>

<html lang="en">

<head>
```

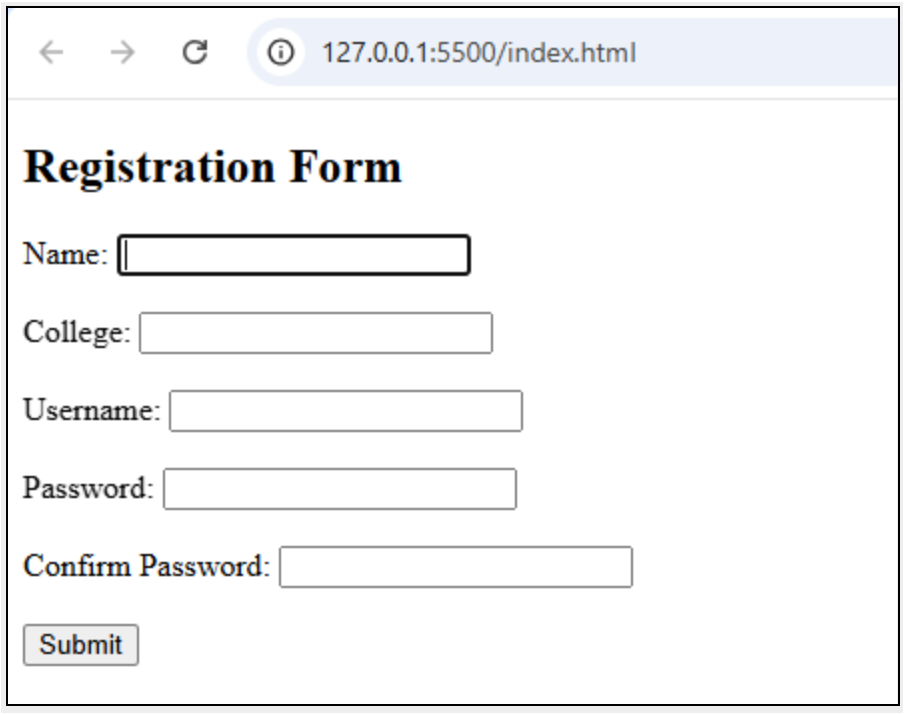
```
<meta charset="UTF-8">
<title>Registration Page</title>
<style>
  body {
    font-family: Arial;
    margin: 50px;
  }
  input, select {
    display: block;
    margin: 10px 0;
    padding: 8px;
    width: 300px;
  }
  #message {
    margin-top: 15px;
    font-weight: bold;
  }
  .error {
    color: red;
  }
  .success {
    color: green;
  }
  datalist {
    width: 300px;
  }
</style>
```

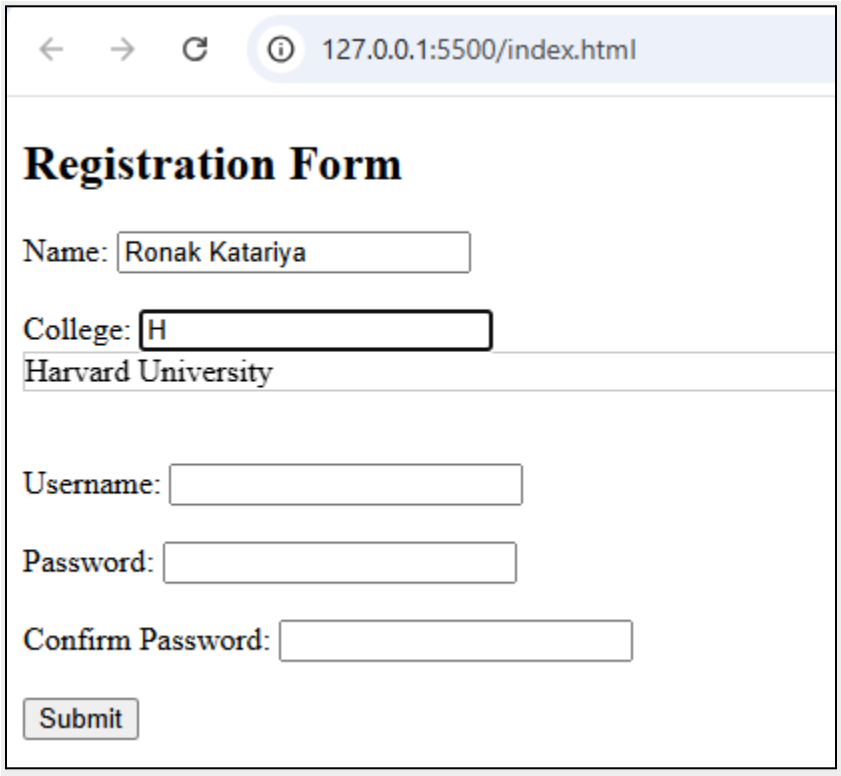
```
</head>
<body>

<h2>Register</h2>

<form id="registerForm">
  <label>Name:</label>
  <input type="text" id="name" required>

  <label>College:</label>
  <input list="colleges" id="college" required>
  <datalist id="colleges">
    <option value="MIT">
    <option value="Stanford University">
    <option value="Harvard University">
    <option value="IIT Delhi">
    <option value="Oxford University">
  </datalist>

  <label>Username:</label>
  <input type="text" id="username" required>

  <label>Password:</label>
  <input type="password" id="password" required>

  <label>Confirm Password:</label>
  <input type="password" id="confirmPassword" required>
```

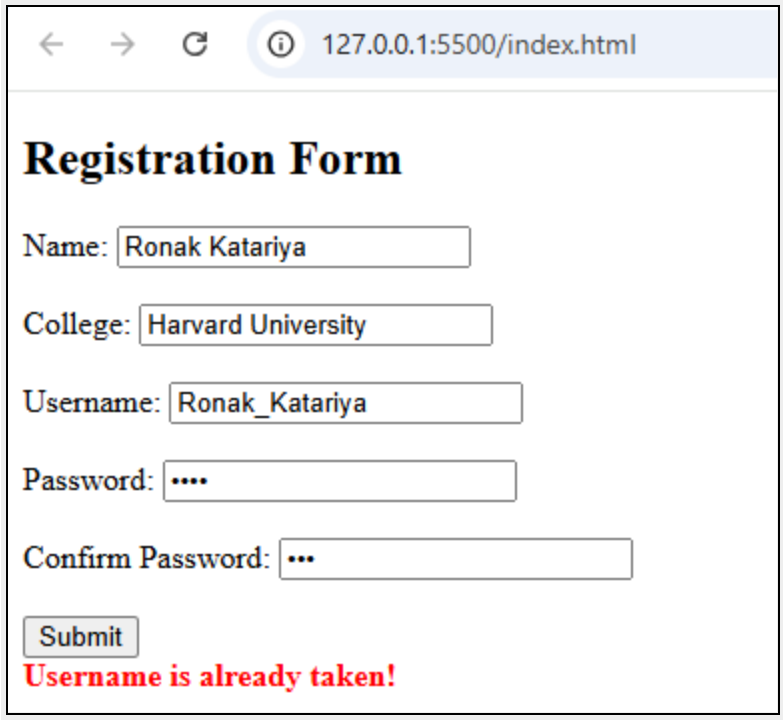
```
  <button type="submit">Register</button>
</form>

<div id="message"></div>

<script>
  // Mock existing usernames
  const existingUsernames = ["john123", "alice99", "mike2024"];

  document.getElementById('registerForm').addEventListener('submit', function(e) {
    e.preventDefault(); // prevent form submission

    // Clear previous message
    const msg = document.getElementById('message');
    msg.innerHTML = '';
    msg.className = '';

    // Get values
    const name = document.getElementById('name').value.trim();
    const college = document.getElementById('college').value.trim();
    const username = document.getElementById('username').value.trim();
    const password = document.getElementById('password').value;
    const confirmPassword = document.getElementById('confirmPassword').value;

    // Validation
    if (name === "") {
```

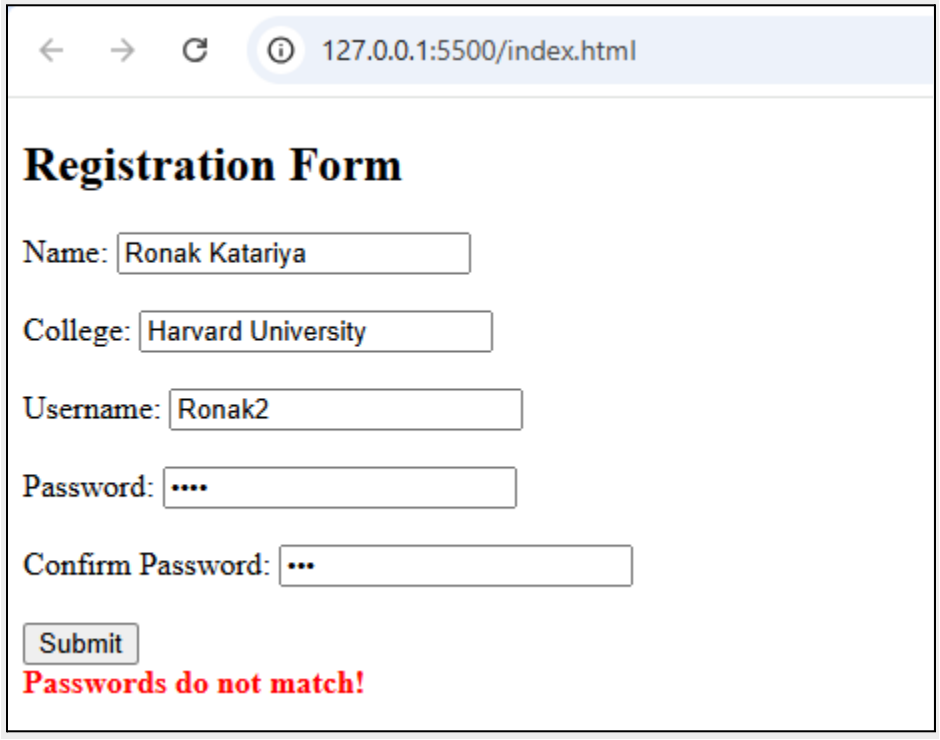
```
  showMessage("Name cannot be empty", "error");
  return;
}
if (existingUsernames.includes(username)) {
  showMessage("Username already exists. Choose a different one.", "error");
  return;
}
if (password !== confirmPassword) {
  showMessage("Passwords do not match", "error");
  return;
}

// Simulate async request using XMLHttpRequest
const xhr = new XMLHttpRequest();
xhr.open("POST", "/register", true); // Fake URL
xhr.setRequestHeader("Content-Type", "application/json;charset=UTF-8");

xhr.onreadystatechange = function() {
  if (xhr.readyState === 4) {
    // Simulate response
    if (xhr.status === 200 || xhr.status === 0) {
      showMessage("Successfully Registered", "success");

      // Simulate adding the new username to the list
      existingUsernames.push(username);
    }
  }
```

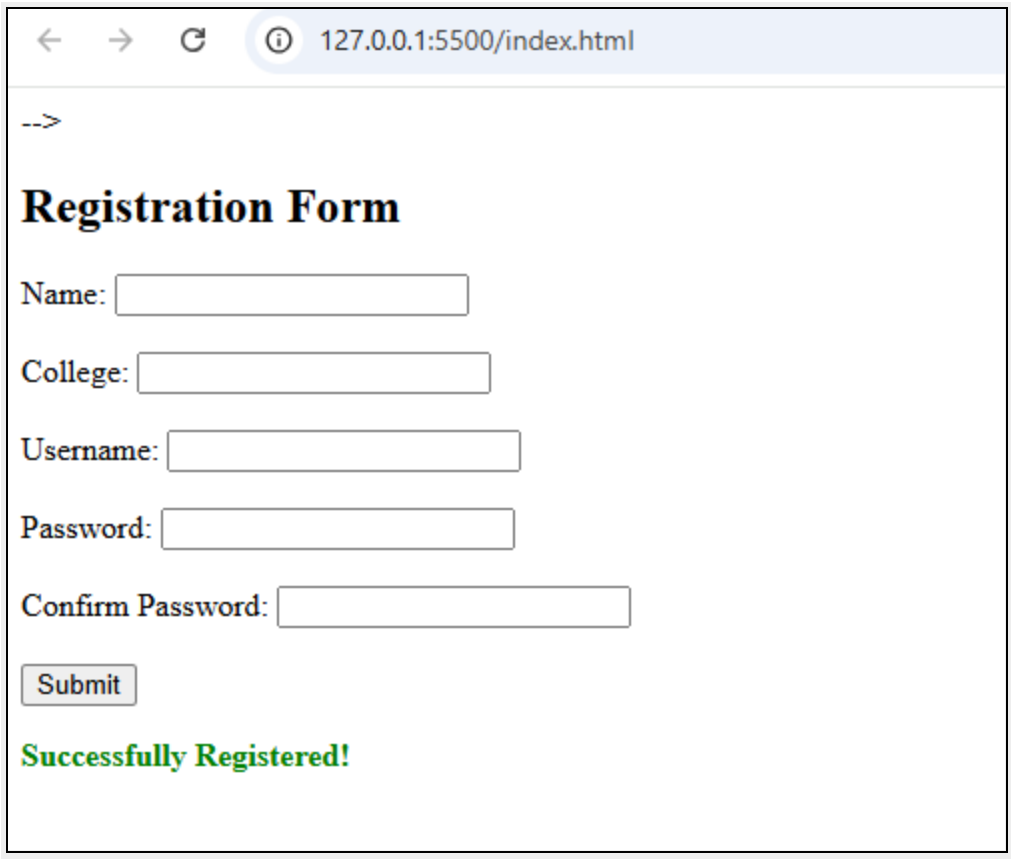
```
    };

    const data = {
      name: name,
      college: college,
      username: username,
      password: password
    };

    xhr.send(JSON.stringify(data));
  });

  function showMessage(text, type) {
    const msg = document.getElementById('message');
    msg.innerHTML = text;
    msg.className = type;
  }
</script>

</body>
</html>
```


127.0.0.1:5500/index.html
Registration Form
Name:
College:
Username:
Password:
Confirm Password:
Submit


127.0.0.1:5500/index.html
Registration Form
Name: Ronak Katariya
College: H
Harvard University
Username:
Password:
Confirm Password:
Submit


127.0.0.1:5500/index.html
Registration Form
Name: Ronak Katariya
College: Harvard University
Username: Ronak_Katariya
Password: ••••
Confirm Password: •••
Submit
Username is already taken!


127.0.0.1:5500/index.html
Registration Form
Name: Ronak Katariya
College: Harvard University
Username: Ronak2
Password: ••••
Confirm Password: •••
Submit
Passwords do not match!


127.0.0.1:5500/index.html
-->
Registration Form
Name:
College:
Username:
Password:
Confirm Password:
Submit
Successfully Registered!