LAB Manual
PART A
(PART A: TO BE REFFERED BY STUDENTS)

Experiment No. 3

A.1 Aim:
To implement CAPTCHA validation in HTML form
Or
To implement user Authentication using any biometric feature
Or
To implement Single Sign on (SSO) system

A.2 Prerequisite:
Understanding of Authentication methods

A.3 Outcome:
After successful completion of this experiment students will be able to
Appreciate the importance of form validation/ biometric authentication

A.4 Theory:

What Is CAPTCHA?

CAPTCHA stands for "Completely Automated Public Turing test to tell Computers and Humans Apart." This term was coined in 2003 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. It's a type of challenge-response test which is used to determine whether the user is human or not.

CAPTCHAs add security to websites by providing challenges that are difficult for bots to perform but relatively easy for humans. For example, identifying all the images of a car from a set of multiple images is difficult for bots but simple enough for human eyes.

The idea of CAPTCHA originates from the Turing Test. A Turing Test is a method to test whether a machine can think like a human or not. Interestingly, a CAPTCHA test can be called a "reverse Turing Test" since in this case, the computer creates the test that challenges humans.

Why Your Website Needs CAPTCHA Validation?

CAPTCHAs are mainly used to prevent bots from automatically submitting forms with spam and other harmful content. Even companies like Google use it to prevent their system from spam attacks. Here are some of the reasons why your website stands to benefit from CAPTCHA validation:

- CAPTCHAs help to prevent hackers and bots from spamming the registration systems by creating fake accounts. If they aren't prevented, they can use those accounts for nefarious purposes.
- CAPTCHAs can forbid brute force log-in attacks from your website which hackers use to try logging in using thousands of passwords.
- CAPTCHAs can restrict bots from spamming the review section by providing false comments.
- CAPTCHAs aid in preventing ticket inflation as some people purchase many tickets for reselling. CAPTCHA can even prevent false registrations to free events.
- CAPTCHAs can restrict cyber crooks from spamming blogs with dodgy comments and links to harmful websites.

Biometric Authentication

Biometric authentication involves using some part of your physical makeup to authenticate you. This could be a fingerprint, an iris scan, a retina scan, or some other physical characteristic. A single characteristic or multiple characteristics could be used. It all depends on the infrastructure and the level of security desired. With biometric authentication, the physical characteristic being examined is usually mapped to a username. This username is used to make decisions after the person has been authenticated.

For more details visit link:
https://heimdalsecurity.com/blog/biometric-authentication/

Single Sign on (SSO)

Authenticating to multiple systems is unpopular with users. Left on their own, users will reuse the same password to avoid having to remember many different passwords. For example, users become frustrated at having to authenticate to a computer, a network, a mail system, an accounting system, and numerous web sites. The panacea for this frustration is called **single sign-on**. A user authenticates once per session, and the system forwards that authenticated identity to all other processes that would require authentication.
https://developers.onelogin.com/saml/python
https://www.miniorange.com/python-adfs-single-sign-on(sso)

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

Roll. No. A022	Name: Kartik Padave
Class: B. Tech CSBS	Batch: 1st
Date of Experiment: 14/01/2022	Date of Submission:
Grade:	

B.1 Software Code written by student:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {
            background-color: white;
        }

        #captchaBackground {
            height: 220px;
            width: 250px;
            background-color: white;
            display: flex;
            align-items: center;
            justify-content: center;
            flex-direction: column;
        }

        #captchaHeading {
            color: black;
        }

        #captcha {
            height: 50%;
            width: 80%;
            font-size: 30px;
            letter-spacing: 3px;
            margin: auto;
            display: block;
            top: 0;
            bottom: 0;
            left: 0;
```

```
            right: 0;
        }

        .center {
            display: flex;
            flex-direction: column;
            align-items: center;
        }

        #submitButton {
            margin-top: 2em;
            margin-bottom: 2em;
            background-color: slategray;
            color: white;
            font-weight: bold;
        }

        #refreshButton {
            background-color: slategray;
            color: white;
            font-weight: bold;
        }

        #textBox {
            height: 25px;
        }

        .incorrectCaptcha {
            color: #FF0000;
        }

        .correctCaptcha {
            color: #7FFF00;
        }
    </style>
    <title>Practical 3 - CAPTCHA Validator</title>
</head>
<body>
    <div class="center">
        <h1 id="captchaHeading">Captcha Validator<h1>
        <div id="captchaBackground">
            <canvas id="captcha">captcha text</canvas>
            <input id="textBox" type="text" name="text">
            <div id="buttons">
                <input id="submitButton" type="submit">
```

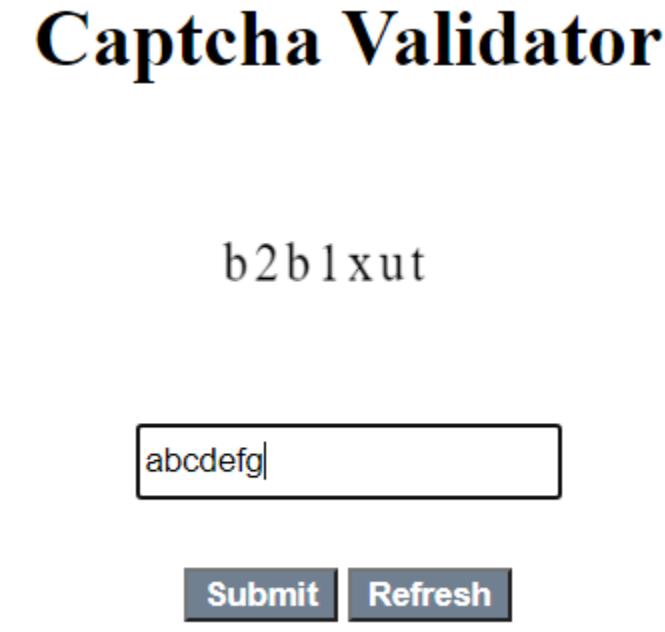
```
                <button id="refreshButton" type="submit">Refresh</button>
            </div>
            <span id="output"></span>
        </div>
    </div>
    <script>
        let captchaText = document.querySelector('#captcha');
        var ctx = captchaText.getContext("2d");
        ctx.font = "30px Roboto";
        ctx.fillStyle = "black";

        let userText = document.querySelector('#textBox');
        let submitButton = document.querySelector('#submitButton');
        let output = document.querySelector('#output');
        let refreshButton = document.querySelector('#refreshButton');

        let alphaNums = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K',
'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a',
'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q',
'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6',
'7', '8', '9'];
        let emptyArr = [];

        for (let i = 1; i <= 7; i++) {
            emptyArr.push(alphaNums[Math.floor(Math.random() *
alphaNums.length)]);
        }

        var c = emptyArr.join('');
        ctx.fillText(emptyArr.join(''),captchaText.width/4,
captchaText.height/2);

        userText.addEventListener('keyup', function(e) {
            if (e.keyCode === 13) {
                if (userText.value === c) {
                    output.classList.add("correctCaptcha");
                    output.innerHTML = "Correct!";
                }
                else {
                    output.classList.add("incorrectCaptcha");
                    output.innerHTML = "Incorrect, please try again";
                }
            }
        });
```

```
        submitButton.addEventListener('click', function() {
            if (userText.value === c) {
                output.classList.add("correctCaptcha");
                output.innerHTML = "Correct!";
            }
            else {
                output.classList.add("incorrectCaptcha");
                output.innerHTML = "Incorrect, please try again";
            }
        });

        refreshButton.addEventListener('click', function() {
            userText.value = "";
            let refreshArr = [];
            for (let j = 1; j <= 7; j++) {
                refreshArr.push(alphaNums[Math.floor(Math.random() *
alphaNums.length)]);
            }

            ctx.clearRect(0, 0, captchaText.width, captchaText.height);
            c = refreshArr.join('');
            ctx.fillText(refreshArr.join(''),captchaText.width/4,
captchaText.height/2);
            output.innerHTML = "";
        });
    </script>
</body>
</html>
```

B.2 Input and Output:
Input:

1. Input with correct captcha

2. Input with wrong captcha



Output:

1. Correct captcha output

2. Wrong captcha output

B.3 Observations and learning:
Users are asked to decode the image and enter the alphanumeric characters in the correct order before submitting the form. Upon form submission, the response is verified, and users are either taken to the next step or presented with an error.

B.4 Conclusion:
Many organizations and businesses have suffered heavy losses like data breaches, spam attacks, etc. because of not having CAPTCHA forms on their websites. It's highly recommended to add CAPTCHA to your website, as it adds a security layer to prevent the website from cybercriminals.
Google also launched a free service called "reCAPTCHA" that helps in protecting websites from spam and abuse. CAPTCHA and reCAPTCHA seem similar, but they're not quite the same thing. Sometimes CAPTCHAs feel frustrating and difficult to understand for many users. Although, there's an important reason as to why they're made to be difficult.

B.5 Questions of Curiosity
(To be answered by student based on the practical performed and learning/observations)

Q1: Discuss any five recent approaches of user authentication

1. Password based authentication
   Passwords are the most common methods of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options. The truth is that there are a lot of passwords to remember. As a result, many people choose convenience over security. Most people use simple passwords instead of creating reliable passwords because they are easier to remember. The bottom line is that passwords have a lot of weaknesses and are not sufficient in protecting online information. Hackers can easily guess user credentials by running through all possible combinations until they find a match.

2. Multi factor authentication
   Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user's smartphone, Captcha tests, fingerprints, voice biometrics or facial recognition.

3. Certificate Based authentication
   Certificate-based authentication technologies identify users, machines, or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver's license or a passport. The certificate contains the digital

identity of a user including a public key, and the digital signature of a certification authority. Digital certificates prove the ownership of a public key and issued only by a certification authority.

4. Biometric based authentication
   Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Here are key advantages of using biometric authentication technologies:
   - Biological characteristics can be easily compared to authorized features saved in a database.
   - Biometric authentication can control physical access when installed on gates and doors.
   - You can add biometrics into your multi-factor authentication process.

5. Token Based authentication
   Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.