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 a large number of 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)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. A016	Name: Varun Mahendra Khadayate
Class B.Tech CsBs	Batch: 1
Date of Experiment: 14-01-2022	Date of Submission: 14-01-2022
Grade:	

B.1 Software Code written by student:

(Paste your Program code completed during the 2 hours of practical in the lab here)

```
<!DOCTYPE html>
<html>
<head>
  <title>Captcha Validation Practical 3</title>
  <script type="text/javascript">
    function GenerateCaptcha() {
      var chr1 = Math.ceil(Math.random() * 10) + '';
      var chr2 = Math.ceil(Math.random() * 10) + '';
      var chr3 = Math.ceil(Math.random() * 10) + '';

      var str = new Array(4).join().replace(/(.|$)/g, function () { return ((Math.random() *
36) | 0).toString(36)[Math.random() < .5 ? "toString" : "toUpperCase"](); });
      var captchaCode = str + chr1 + ' ' + chr2 + ' ' + chr3;
      document.getElementById("txtCaptcha").value = captchaCode
    }

    function ValidCaptcha() {
      var str1 = removeSpaces(document.getElementById('txtCaptcha').value);
      var str2 = removeSpaces(document.getElementById('txtCompare').value);

      if (str1 == str2) return true;
      return false;
    }

    function removeSpaces(string) {
      return string.split(' ').join('');
    }
  </script>

</head>
```

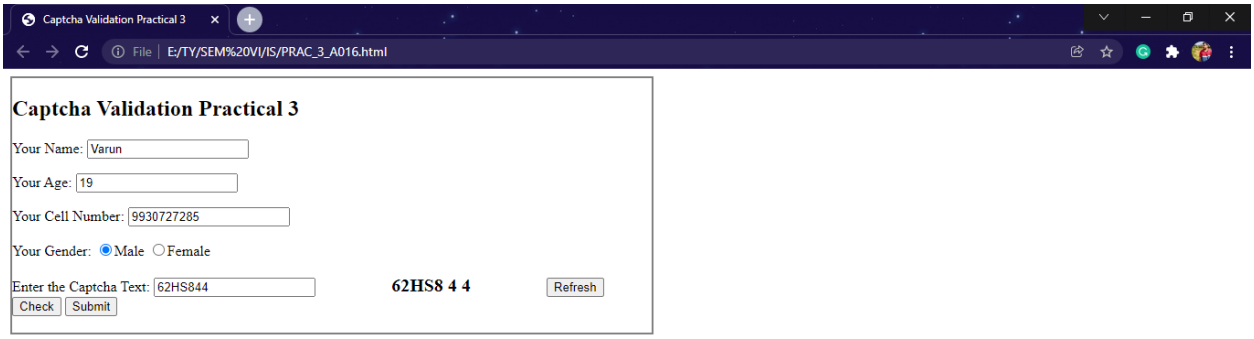
```
<body onload="GenerateCaptcha();">
  <div style="border: 2px solid gray; width: 700px;">
    <h2>Captcha Validation Practical 3</h2>
    <form method="post" action="">
      <p>Your Name:
        <input type="text" name="textified">
      </p>
      <p>Your Age:
        <input type="text" name="textified">
      </p>
      <p>Your Cell Number:
        <input type="text" name="textified">
      </p>
      <p>Your Gender:
        <input type="radio" name="gender" value="m">Male
        <input type="radio" name="gender" value="f">Female
      </p>
    </form>
    Enter the Captcha Text:
    <input type="text" id="txtCompare" />
    <input type="text" id="txtCaptcha" style="text-align: center; border: none; font-
weight: bold; font-size: 20px; font-family: Modern" />
    <input type="button" id="btnrefresh" value="Refresh" onclick="GenerateCaptcha();"
/>
    <input id="btnValid" type="button" value="Check" onclick="alert(ValidCaptcha());"
/>
    <button type="submit" form="form1" value="Submit">Submit</button>
    <br />
    <br />
  </div>
</body>
</html>
```

B.2 Input and Output:

(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)

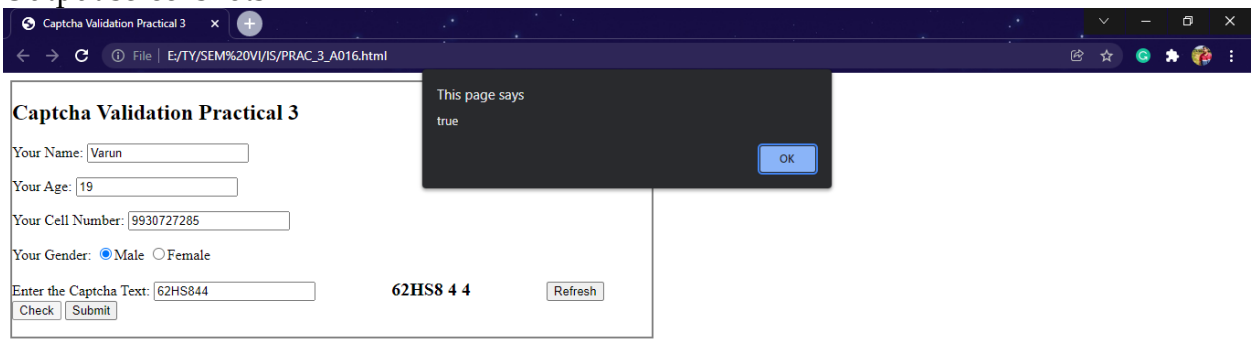
Input:

1. Input string acting as password/captcha or biometric feature



Output:

Output screenshots



B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

We were able to perform the experiment with the use of **JavaScript** and **HTML** and were able to get the desired output needed.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Hence, we were able to implement CAPTCHA validation in HTML form.

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.

However, passwords are prone to phishing attacks and bad hygiene that weakens effectiveness. An average person has about 25 different online accounts, but only 54% of users use different passwords across their accounts.

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.

MFA authentication methods and technologies increase the confidence of users by adding multiple layers of security. MFA may be a good defense against most account hacks, but it has its own pitfalls. People may lose their phones or SIM cards and not be able to generate an authentication code.

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.

Users provide their digital certificates when they sign into a server. The server verifies the credibility of the digital signature and the certificate authority. The server then uses cryptography to confirm that the user has a correct private key associated with the certificate.

4. Biometric 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.

Biometric authentication technologies are used by consumers, governments and private corporations including airports, military bases, and national borders. The technology is increasingly adopted due to the ability to achieve a high level of security without creating friction for the user. Common biometric authentication methods include:

- **Facial recognition**—matches the different face characteristics of an individual trying to gain access to an approved face stored in a database. Face recognition can be inconsistent when comparing faces at different angles or comparing people who look similar, like close relatives. Facial liveness like ID R&D's passive facial liveness prevents spoofing.
- **Fingerprint scanners**—match the unique patterns on an individual's fingerprints. Some new versions of fingerprint scanners can even assess the vascular patterns in people's fingers. Fingerprint scanners are currently the most popular biometric technology for everyday consumers, despite their frequent inaccuracies. This popularity can be attributed to iPhones.
- **Speaker Recognition** —also known as voice biometrics, examines a speaker's speech patterns for the formation of specific shapes and sound qualities. A voice-protected device usually relies on standardized words to identify users, just like a password.
- **Eye scanners**—include technologies like iris recognition and retina scanners. Iris scanners project a bright light towards the eye and search for unique patterns in the colored ring around the pupil of the eye. The patterns are then compared to approved information stored in a database. Eye-based authentication may suffer inaccuracies if a person wears glasses or contact lenses.

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.