Batch: C1 Roll No.: 16010122323
Student Name: Vedansh Savla.
Experiment No: 03
Staff In-charge: Shivani Deosthale

TITLE: Implementation of CAPTCHA for Security of systems
AIM: To implement Text based, Audio based, Image based, Mathematical based CAPTCHA. without using inbuilt functions of python or any such programming language.
OUTCOME: Student will be able to
CO4: Illustrate and Compare network security mechanisms

Theory about Network Security and role of CAPTCHA:

Network Security involves protecting systems and data from unauthorized access or attacks through measures like firewalls, encryption, and intrusion detection. It aims to safeguard information and ensure secure operations across networks.

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a tool used to distinguish between human users and bots, preventing automated systems from abusing websites. It plays a key role in network security by protecting against spam, brute-force attacks, and bot-driven misuse.

Types of CAPTCHA:

- **Text-based**: Distorted letters and numbers.
- **Image-based**: Users select matching images (e.g., reCAPTCHA).
- **Audio-based**: Spoken characters for visually impaired users.
- **Checkbox CAPTCHA**: A simple "I'm not a robot" checkbox.
- **Invisible CAPTCHA**: Tracks user behavior to detect bots.

Benefits: CAPTCHA prevents automated attacks, protects sensitive data, reduces fake accounts, and ensures fair access to services.

Limitations: Some CAPTCHAs are not accessible for people with disabilities, advanced bots can bypass them, and they may raise privacy concerns.

Algorithm:

Step 1: Generate CAPTCHA (Visual)

1. **Initialize** a `captchaCode` variable to store the randomly generated CAPTCHA code.
2. **Generate a Random CAPTCHA Code**:
   - Choose a random length (between 5 and 10 characters).
   - Randomly choose between numbers (0-9) and uppercase letters (A-Z).
3. **Draw CAPTCHA Code on Canvas**:
   - Create a canvas element and set its width and height.
   - Add random lines and distortions to the background.
   - Draw each character of the generated CAPTCHA on the canvas with random positions and rotation.
4. **Save the CAPTCHA Code** in a variable (`captchaCode`) for later validation.

Step 2: Generate and Display Image CAPTCHA

1. **Define Image Data**:
   - Create an array of image objects with `src` (image path), `alt` (image description), and `index` (image identifier).
2. **Shuffle Images**:
   - Shuffle the array of images using the Fisher-Yates algorithm to ensure randomness.
3. **Display Images on the Page**:
   - Dynamically generate `<img>` tags for each image and display them in a container (`captchaImageContainer`).
   - Each image has an `onclick` event listener to toggle selection when clicked.

Step 3: Select Images for Image CAPTCHA

1. **Track Selected Images**:
   - Create an empty array `selectedImages` to track which images have been selected.
2. **Toggle Image Selection**:

- When an image is clicked, check if it’s already selected:
    - If selected, remove it from the `selectedImages` array and update the CSS to unselect it.
    - If not selected, add it to the `selectedImages` array and update the CSS to highlight it.

Step 4: Handle Audio CAPTCHA

1. **Generate Audio CAPTCHA**:
    - On button click (for audio CAPTCHA), read out each character of the generated CAPTCHA using the Web Speech API.
2. **Speech Synthesis**:
    - Use the `SpeechSynthesisUtterance` API to speak each character of the CAPTCHA, one by one.
    - Each character is spoken with a slight delay before the next character is read aloud.

Step 5: Submit Form (Login Validation)

1. **On Form Submit**:
    - **Prevent Default Form Submission**: Use `event.preventDefault()` to prevent the form from being submitted traditionally.
    - Retrieve the input values: `username`, `password`, and `captchaInput` (for visual CAPTCHA).
2. **Check Username and Password**:
    - If the username is "admin" and the password is "password":
        - Validate the CAPTCHA:
            - Check if the entered `captchaInput` matches the `captchaCode` (visual CAPTCHA).
            - Check if the selected images match the correct images (for the image CAPTCHA).
        - If both validations pass:
            - Display a success message and **clear the CAPTCHA input**.
        - If CAPTCHA fails:
            - Display an error message for invalid CAPTCHA.
3. **Invalid Username/Password**:
    - If the username or password is incorrect, display an error message for invalid login.

Implementation:

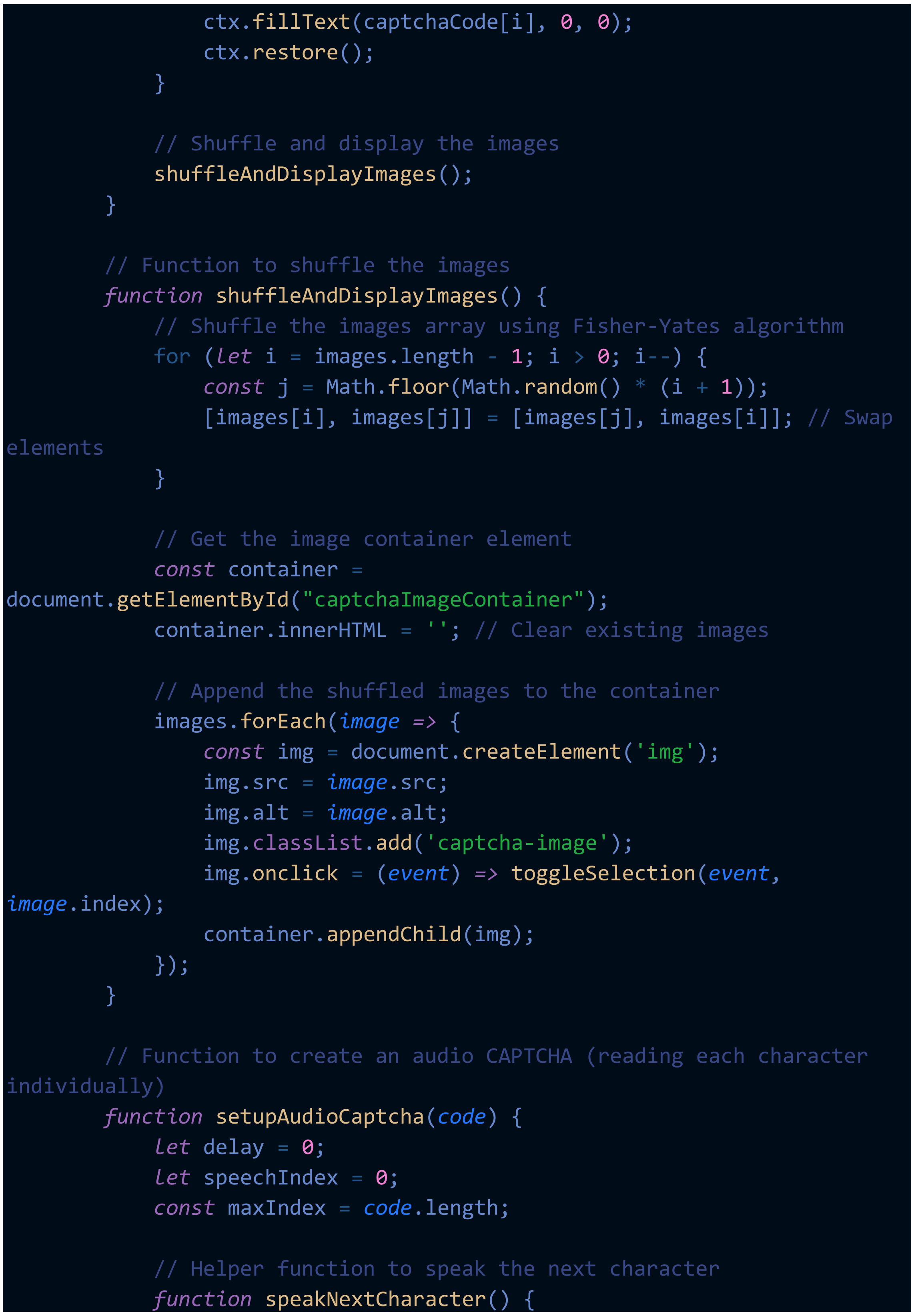
1. Implement Text based, Audio based, Image based, Mathematical based CAPTCHA, re-CAPTCHA. without using inbuilt functions of python or any such programming language.

Learning Dialog - YouTube video:

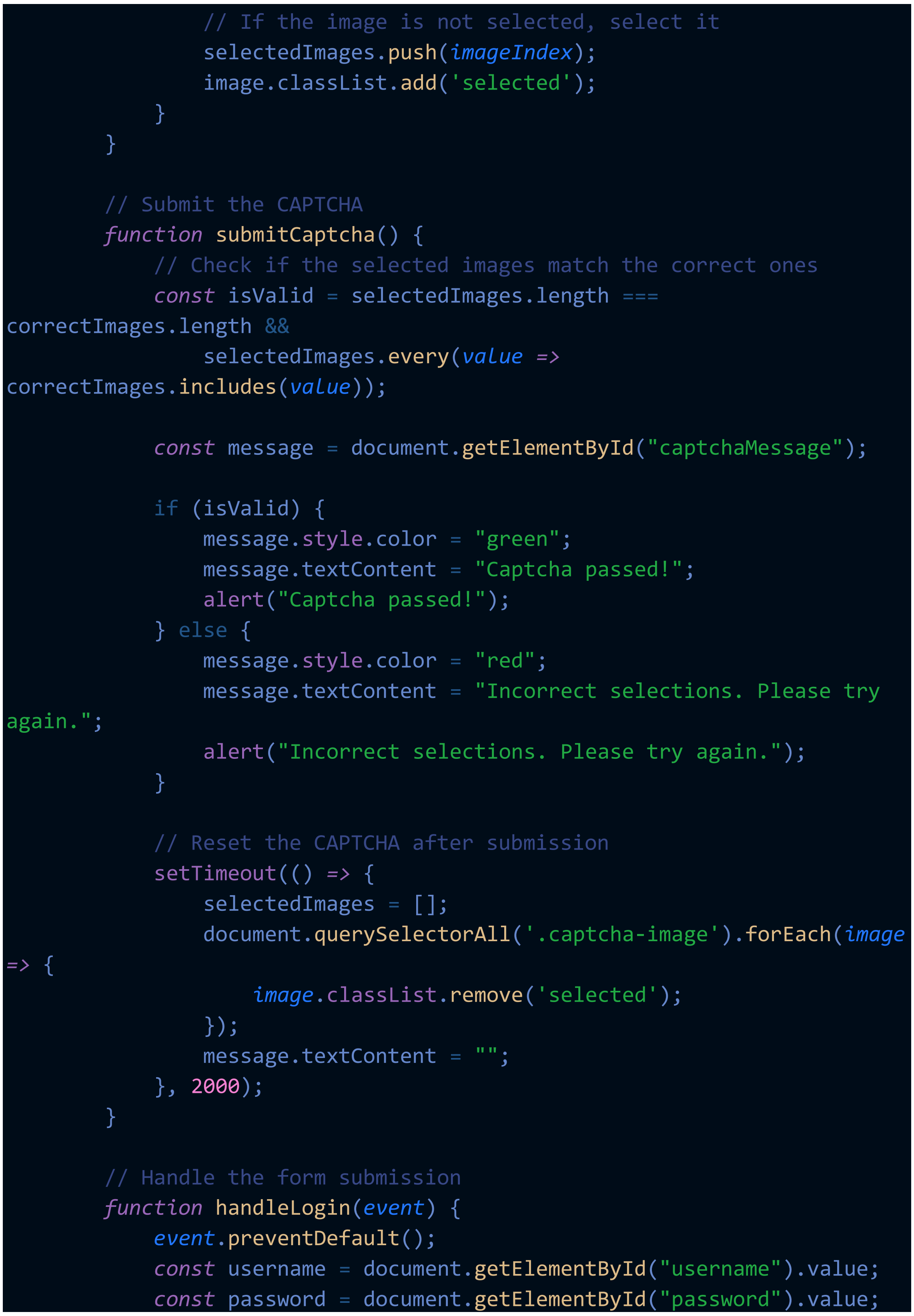
https://www.youtube.com/watch?v=bfKwizfuuOU

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Login Page with Captcha</title>
    <style>
        .captcha-images {
            display: flex;
            flex-wrap: wrap;
            gap: 10px;
        }

        .captcha-image {
            width: 150px;
            height: 150px;
            object-fit: cover;
            border: 2px solid #ccc;
            cursor: pointer;
        }

        .selected {
            border: 2px solid green;
        }

        .captcha-message {
            color: red;
            font-size: 16px;
        }
    </style>
</head>
```

```
<body>
    <h1>Login</h1>
    <form onsubmit="return handleLogin(event)">
        <label for="username">Username:</label>
        <input type="text" id="username" required>
        <br>
        <label for="password">Password:</label>
        <input type="password" id="password" required>
        <hr>

        <!-- Visual CAPTCHA (Canvas) -->
        <canvas id="captchaCanvas" width="200" height="50"></canvas>
        <br>
        <input type="text" id="captchaInput" placeholder="Enter Captcha"
required>
        <hr>

        <!-- Image CAPTCHA -->
        <p>Select all images with fast food logos:</p>
        <div class="captcha-images" id="captchaImageContainer">
            <!-- Images will be inserted here dynamically -->
        </div>
        <hr>

        <button type="button" onclick="generateCaptcha()">Refresh
Captcha</button>
        <button type="button" onclick="playAudioCaptcha()">Play Audio
Captcha</button>
        <button type="submit">Login</button>
    </form>
    <p id="message"></p>

    <script>
        let captchaCode = "";
        let selectedImages = [];
        const correctImages = [1, 3, 5]; // Correct images that contain
the object (e.g., traffic lights)

        const images = [
            { src: 'image1.png', alt: 'Image 1', index: 1 },
            { src: 'image2.png', alt: 'Image 2', index: 2 },
            { src: 'image3.png', alt: 'Image 3', index: 3 },
            { src: 'image4.png', alt: 'Image 4', index: 4 },
            { src: 'image5.png', alt: 'Image 5', index: 5 },
            { src: 'image6.png', alt: 'Image 6', index: 6 }
        ];

        // Generate the CAPTCHA code (visual and audio)
        function generateCaptcha() {
            const canvas = document.getElementById("captchaCanvas");
            const ctx = canvas.getContext("2d");
            ctx.clearRect(0, 0, canvas.width, canvas.height);

            captchaCode = "";
            let r1 = Math.floor(Math.random() * 6) + 5;

            for (let i = 0; i < r1; i++) {
                let r2 = Math.floor(Math.random() * 10) + 1;
                if (r2 < 6) {
                    captchaCode += Math.floor(Math.random() * 10);
                } else {
                    captchaCode += String.fromCharCode(65 +
Math.floor(Math.random() * 26));
                }
            }

            ctx.fillStyle = '#f0f0f0';
            ctx.fillRect(0, 0, canvas.width, canvas.height);

            for (let i = 0; i < 5; i++) {
                ctx.beginPath();
                ctx.moveTo(Math.random() * canvas.width, Math.random() *
canvas.height);
                ctx.lineTo(Math.random() * canvas.width, Math.random() *
canvas.height);
                ctx.stroke();
            }

            ctx.font = '24px Times New Roman';
            ctx.fillStyle = 'green';
            for (let i = 0; i < captchaCode.length; i++) {
                let y = Math.random() * 10 + 25;
                let rotation = (Math.random() * 0.4 - 0.2);
                ctx.save();
                ctx.translate(i * 20, y);
                ctx.rotate(rotation);
```

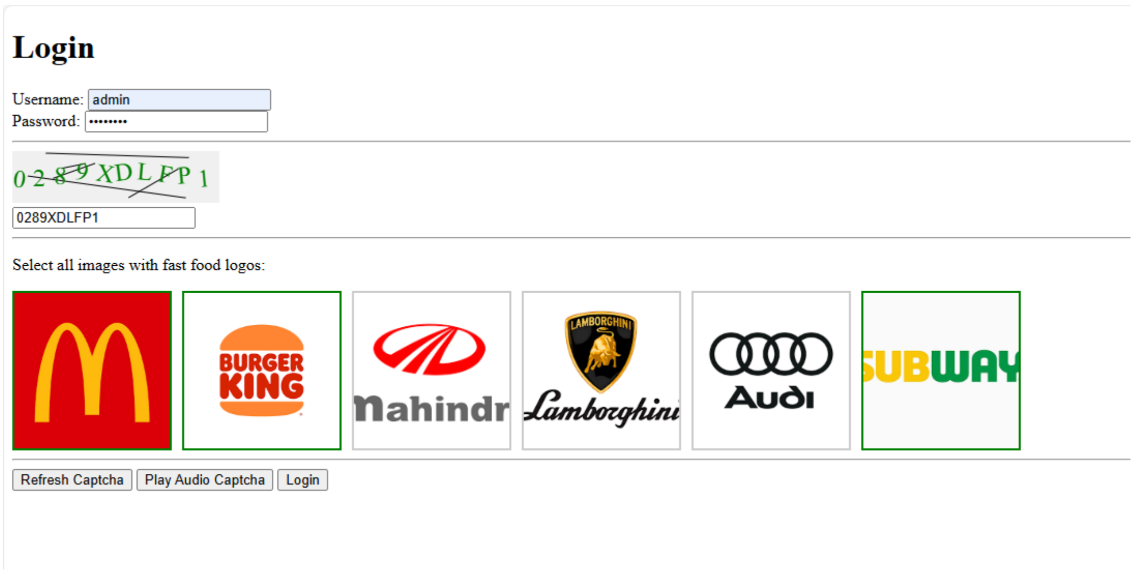


```
                ctx.fillText(captchaCode[i], 0, 0);
                ctx.restore();
            }

            // Shuffle and display the images
            shuffleAndDisplayImages();
        }

        // Function to shuffle the images
        function shuffleAndDisplayImages() {
            // Shuffle the images array using Fisher-Yates algorithm
            for (let i = images.length - 1; i > 0; i--) {
                const j = Math.floor(Math.random() * (i + 1));
                [images[i], images[j]] = [images[j], images[i]]; // Swap
elements
            }

            // Get the image container element
            const container =
document.getElementById("captchaImageContainer");
            container.innerHTML = ''; // Clear existing images

            // Append the shuffled images to the container
            images.forEach(image => {
                const img = document.createElement('img');
                img.src = image.src;
                img.alt = image.alt;
                img.classList.add('captcha-image');
                img.onclick = (event) => toggleSelection(event,
image.index);
                container.appendChild(img);
            });
        }

        // Function to create an audio CAPTCHA (reading each character
individually)
        function setupAudioCaptcha(code) {
            let delay = 0;
            let speechIndex = 0;
            const maxIndex = code.length;

            // Helper function to speak the next character
            function speakNextCharacter() {
                if (speechIndex >= maxIndex) return;  // Stop when all characters are read

                const char = code[speechIndex];
                const speech = new SpeechSynthesisUtterance(char);
                speech.lang = 'en-US';
                speech.rate = 0.75; // Slow down the speech (default is 1)
                speech.voice = window.speechSynthesis.getVoices().find(voice => voice.name === 'Google UK English Male');
                speech.pitch = 1; // Adjust pitch if needed

                // When a speech finishes, move to the next character
                speech.onend = function () {
                    speechIndex++; // Move to the next character
                    setTimeout(speakNextCharacter, 200);  // Wait for a bit before speaking the next character
                };

                window.speechSynthesis.speak(speech);  // Start speaking the current character
            }

            // Start speaking the first character
            speakNextCharacter();
        }

        // Function to play the audio CAPTCHA when the button is clicked
        function playAudioCaptcha() {
            setupAudioCaptcha(captchaCode);
        }

        // Toggle selection of images for image CAPTCHA
        function toggleSelection(event, imageIndex) {
            const image = event.target;
            const index = selectedImages.indexOf(imageIndex);

            if (index > -1) {
                // If the image is already selected, unselect it
                selectedImages.splice(index, 1);
                image.classList.remove('selected');
            } else {
```



```
                // If the image is not selected, select it
                selectedImages.push(imageIndex);
                image.classList.add('selected');
            }
        }

        // Submit the CAPTCHA
        function submitCaptcha() {
            // Check if the selected images match the correct ones
            const isValid = selectedImages.length ===
correctImages.length &&
                selectedImages.every(value =>
correctImages.includes(value));

            const message = document.getElementById("captchaMessage");

            if (isValid) {
                message.style.color = "green";
                message.textContent = "Captcha passed!";
                alert("Captcha passed!");
            } else {
                message.style.color = "red";
                message.textContent = "Incorrect selections. Please try
again.";
                alert("Incorrect selections. Please try again.");
            }

            // Reset the CAPTCHA after submission
            setTimeout(() => {
                selectedImages = [];
                document.querySelectorAll('.captcha-image').forEach(image
=> {
                    image.classList.remove('selected');
                });
                message.textContent = "";
            }, 2000);
        }

        // Handle the form submission
        function handleLogin(event) {
            event.preventDefault();
            const username = document.getElementById("username").value;
            const password = document.getElementById("password").value;
            const captchaInput =
document.getElementById("captchaInput").value;

            if (username === "admin" && password === "password") {
                // Check for both visual and image CAPTCHA validity
                if (captchaCode === captchaInput && selectedImages.length
=== correctImages.length &&
                    selectedImages.every(value =>
correctImages.includes(value))) {
                    // Clear input fields only on successful login
                    document.getElementById("captchaInput").value = "";
                    alert("Login successful!");
                    generateCaptcha();
                } else {
                    alert("Invalid captcha!!!");
                    generateCaptcha();
                }
            } else {
                alert("Invalid login!!!");
                generateCaptcha();
            }
        }

        // Generate the captcha when the page loads
        window.onload = generateCaptcha;
    </script>
</body>

</html>
```

Output:


Login

Username: admin

Password: ••••••••

0289XDLFP1

0289XDLFP1

Select all images with fast food logos:

Post Lab Questions:

1. Explore other forms / types of CAPTCHAs

=> CAPTCHAs are used to differentiate humans from bots. Here are some types:

1. **Checkbox CAPTCHA:** A user checks a box to confirm they are human.

Drawback: Bots can mimic human interactions.

2. **Puzzle CAPTCHA:** Users solve a simple puzzle. Drawback: Difficult for users with motor impairments.
3. **Behavior-based CAPTCHA (e.g., reCAPTCHA v3):** Analyzes user behavior without interaction. Drawback: Raises privacy concerns and can be mimicked by sophisticated bots.
4. **Biometric CAPTCHA:** Uses facial recognition or fingerprints for identification. Drawback: Requires special hardware and raises privacy concerns.
5. **Math-based CAPTCHA:** Users solve a simple math problem. Drawback: Bots can easily solve simple problems.

2. Write limitations of CAPTCHA.

=> While CAPTCHAs block bots, they have limitations:

- Usability Issues: Difficult for users with disabilities (e.g., vision or hearing impairments).
- Accessibility: Not all CAPTCHAs are fully accessible; audio alternatives may not always be effective.
- User Experience: CAPTCHAs can disrupt the user experience and be frustrating.

- False Positives: CAPTCHAs can incorrectly flag legitimate users as bots.
- Evasion by Advanced Bots: Bots are becoming more sophisticated and can bypass CAPTCHAs.
- Privacy Concerns: Behavior-based and biometric CAPTCHAs may raise privacy issues.
- Inconvenience: CAPTCHAs may lead to abandoned forms or transactions, especially for slow or limited devices.
- Impact on SEO: Some CAPTCHAs may prevent web crawlers from indexing content, affecting SEO.

Conclusion: