DOCUMENTATION

How To Implement Server Side Validation

(Complete the Story CAPTCHA)

Introduction

This document explains how to perform server-side CAPTCHA validation to ensure that the user is a human and not a bot.

2. Validation Flow

1. The user selects a CAPTCHA answer on the client side.
2. The data is sent to the server using the HTTP POST method.
3. The server validates the answer by comparing the user's input with the expected value.
4. The server returns a response, indicating whether the CAPTCHA was passed or failed.
5. If the user fails too many times, the server may impose a time penalty before allowing another

Steps to Implement Server-Side CAPTCHA Validation

Server Code (server.js)

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
const PORT = 3000;

app.use(cors());
app.use(bodyParser.json());

// Correct answer
const correctAnswer = "Pohon";
```

```
// Temporary storage for tracking failures
let failedAttempts = {};

app.post('/validate-captcha', (req, res) => {
    const { sessionId, answer } = req.body;

    if (!sessionId) {
        return res.status(400).json({ success: false, message:
"Session ID is required." });
    }

    // Initialize failure count if not already set
    if (!failedAttempts[sessionId]) {
        failedAttempts[sessionId] = 0;
    }

    // Check if the user has exceeded the failure limit
    if (failedAttempts[sessionId] >= 3) {
        return res.status(403).json({ success: false, message:
"You are locked out. Try again in 15 seconds." });
    }

    // Validate answer
    if (answer === correctAnswer) {
        failedAttempts[sessionId] = 0; // Reset failure count
        return res.json({ success: true, message: "CAPTCHA
passed." });
    } else {
        failedAttempts[sessionId]++;
        return res.json({ success: false, message: "Incorrect
answer. Try again." });
    }
});

app.listen(PORT, () => {
    console.log(`Server running at http://localhost:${PORT}`);
});
```

Client-Side Integration

```
function sendCaptchaAnswer(answer) {
    const sessionId = localStorage.getItem("sessionId") ||
Math.random().toString(36).substr(2, 9);
    localStorage.setItem("sessionId", sessionId);

    fetch('http://localhost:3000/validate-captcha', {
```

```
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ sessionId, answer })
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            alert("CAPTCHA passed!");
            window.location.href =
"https://mathewsin.github.io/CaptchaTester/";
        } else {
            alert(data.message);
        }
    })
    .catch(error => console.error("Error:", error));
     }
```

Conclusion

- Validation is performed on the server to prevent client-side manipulation.
- Session ID is used to track each user's status uniquely.
- A failure limit (3 attempts) is enforced for additional security.
- Users who fail 3 times will be locked out for 15 seconds.