Here's a reference guide for implementing reCAPTCHA in an application:

1. Obtain API keys: The first step is to obtain API keys for your application from Google's reCAPTCHA website. You will need to register your application and provide the necessary information. Once you have the API keys, you can start implementing reCAPTCHA.
2. Include reCAPTCHA in your application: You will need to add the reCAPTCHA JavaScript API to your application. This is done by including the following code in the head section of your HTML file:

htmlCopy code

**<script src="https://www.google.com/recaptcha/api.js" async defer></script>**

1. Add the reCAPTCHA widget to your form: Next, you need to add the reCAPTCHA widget to your form. You can do this by adding the following HTML code to your form:

htmlCopy code

**<div class="g-recaptcha" data-sitekey="your\_site\_key"></div>**

Make sure to replace "your\_site\_key" with the site key you obtained from Google.

1. Verify the user's response: When the user submits the form, you need to verify the user's response to the reCAPTCHA challenge. This is done by sending a request to Google's reCAPTCHA server with the user's response and your secret key. The response will include a success or failure status.

Here's an example of how to implement reCAPTCHA in a JavaScript form:

First, you'll need to include the reCAPTCHA API script in your HTML file:

**<!-- Load the reCAPTCHA API script -->**

**<script src="https://www.google.com/recaptcha/api.js" async defer></script>**

Then, you can add the reCAPTCHA widget to your form:

**<!-- Add the reCAPTCHA widget to your form -->**

**<form action="/submit-form" method="post">**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name" required>**

**<label for="email">Email:</label>**

**<input type="email" id="email" name="email" required>**

**<!-- Add the reCAPTCHA widget -->**

**<div class="g-recaptcha" data-sitekey="YOUR\_SITE\_KEY"></div>**

**<button type="submit">Submit</button>**

**</form**>

In the code above, replace "YOUR\_SITE\_KEY" with your reCAPTCHA site key. You can get your site key by registering your site with Google reCAPTCHA.

Finally, you'll need to verify the reCAPTCHA response on the server side. Here's an example of how to do this in Node.js:

**const fetch = require("node-fetch");**

**async function verifyRecaptcha(token) {**

**const secret = "YOUR\_SECRET\_KEY";**

**const response = await fetch(**

**`https://www.google.com/recaptcha/api/siteverify?secret=${secret}&response=${token}`,**

**{ method: "POST" }**

**);**

**const data = await response.json();**

**return data.success;**

**}**

**app.post("/submit-form", async (req, res) => {**

**const { name, email, "g-recaptcha-response": recaptchaToken } = req.body;**

**const isHuman = await verifyRecaptcha(recaptchaToken);**

**if (!isHuman) {**

**res.status(400).send("reCAPTCHA verification failed");**

**return;**

**}**

**// process the form submission**

**res.send("Form submitted successfully!");**

**});**

In the code above, replace "YOUR\_SECRET\_KEY" with your reCAPTCHA secret key.The verifyRecaptcha function sends a request to the reCAPTCHA API to verify the user's response token. The server checks the response from the reCAPTCHA API and processes the form submission if the response is valid.