Autenticacion TOTP

NODE Js

WEB: https://nodejs.org/en
github: https://github.com/Ubyquit/seguridad-us-2024

Paso 1: Instalación de Librerías

Primero, necesitarás instalar las librerías necesarias. Abre tu terminal y ejecuta los siguientes comandos:

```
npm init -y
npm install express mysql speakeasy qrcode body-parser
```

Paso 2: Configuración de la Base de Datos en MySQL

Asegúrate de tener MySQL instalado en tu sistema y luego ejecuta estos comandos en tu cliente MySQL o en una herramienta como phpMyAdmin para crear la base de datos y la tabla necesaria.

Creación de la Base de Datos y Tabla en MySQL:

```
CREATE DATABASE IF NOT EXISTS autenticacion_multifactorial;
USE autenticacion_multifactorial;

CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    secret VARCHAR(50) NOT NULL,
    email VARCHAR(100)
);
```

Paso 3: Desarrollo de la Aplicación en Node.js

Crea un archivo `app.js` en tu directorio de trabajo y copia el siguiente código:

`app.js`:

```
const express = require('express');
const mysql = require('mysql');
const speakeasy = require('speakeasy');
const qrcode = require('qrcode');
const bodyParser = require('body-parser');

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

// Configuración de la conexión a MySQL
const conexion = mysql.createConnection({
  host: 'localhost',
  user: 'tu_usuario',
  password: 'tu_contraseña',
  database: 'autenticacion_multifactorial'
});

// Conexión a MySQL
conexion.connect((err) => {
  if (err) {
    console.error('Error de conexión:', err);
    return;
  }
  console.log('Conexión a MySQL establecida.');
});

// Middleware para procesar JSON y formularios
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

// Ruta para generar código QR y mostrar formulario de registro
app.get('/', (req, res) => {
  res.send(`
    <h1>Registro de Usuario</h1>
```

```
    <form action="/" method="post">
      <label for="username">Nombre de Usuario:</label><br>
      <input type="text" id="username" name="username" required><br><br>
      <input type="submit" value="Registrar">
    </form>
  `);
});

// Ruta para manejar el registro de usuario
app.post('/', (req, res) => {
  const { username } = req.body;

  // Generar clave secreta para el usuario
  const secret = speakeasy.generateSecret({ length: 20 });

  // Generar código QR con la clave secreta
  qrcode.toDataURL(secret.otpauth_url, (err, imageUrl) => {
    if (err) {
      console.error('Error al generar el código QR:', err);
      res.send('Error al generar el código QR.');
      return;
    }

    // Insertar usuario y su clave secreta en la base de datos
    const usuario = { username: username, secret: secret.base32 };
    conexion.query('INSERT INTO users SET ?', usuario, (error, resultados) => {
      if (error) {
        console.error('Error al insertar usuario en la base de datos:', error);
        res.send('Error al insertar usuario en la base de datos.');
        return;
      }

      // Mostrar página con código QR y formulario de autenticación
      res.send(`
        <h1>Escanea este código QR con Google Authenticator</h1>
        <img src="${imageUrl}" alt="QR Code"><br><br>
        <form action="/verificar" method="post">
          <label for="token">Ingresa el código de Google Authenticator:</label><br>
          <input type="text" id="token" name="token" required><br><br>
          <input type="hidden" id="username" name="username" value="${username}">
          <input type="submit" value="Verificar">
        </form>
      `);
    });
  });
});

// Ruta para verificar el código TOTP
app.post('/verificar', (req, res) => {
  const { token, username } = req.body;

  // Consultar la clave secreta del usuario desde MySQL
  const consulta = `SELECT secret FROM users WHERE username = '${username}'`;
  conexion.query(consulta, (error, resultados, campos) => {
    if (error) {
      console.error('Error al consultar la base de datos:', error);
      res.send("Error al verificar la autenticación.");
      return;
    }

    if (resultados.length > 0) {
      const secret = resultados[0].secret;
      const esValido

 = speakeasy.totp.verify({
        secret: secret,
        encoding: 'base32',
        token: token,
      });

      if (esValido) {
        res.send("Código TOTP válido. Autenticación exitosa.");
      } else {
        res.send("Código TOTP inválido. Autenticación fallida.");
      }
    } else {
```

```
      res.send("Usuario no encontrado.");
    }
  });
});

// Iniciar servidor
app.listen(puerto, () => {
  console.log(`Servidor en ejecución en http://localhost:${puerto}`);
});
```

Uso:

1. Guarda el código en un archivo llamado `app.js`.
2. Asegúrate de tener una base de datos MySQL con la tabla `users` creada según lo mencionado en los pasos anteriores.
3. Ejecuta la aplicación Node.js con el comando `node app.js`.
4. Visita `http://localhost:3000/` en tu navegador para registrar un nuevo usuario.
5. Ingresa un nombre de usuario en el formulario y envíalo.
6. Escanea el código QR generado con Google Authenticator en tu dispositivo móvil.
7. Ingresa el código de Google Authenticator y el nombre de usuario en el formulario de verificación y envíalo.
8. La aplicación verificará el código y mostrará un mensaje de autenticación exitosa o fallida.

Ahora el formulario solicitará el nombre de usuario y lo utilizará en todo el proceso de registro y autenticación multifactorial. Recuerda reemplazar `'tu_usuario'` y `'tu_contraseña'` con las credenciales adecuadas de tu base de datos MySQL.