# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 11

Date Of Performance: 09/10/2025

Aim: Implementation of Nodejs Server side Async Call back Programming

CODE (package.json):

{

    "name": "nigga-chan",

    "version": "1.0.0",

    "description": "Node.js Async Callback Programming Demonstration",

    "main": "main.js",

    "scripts": {

        "start": "node main.js",

        "test": "echo \"Run: node main.js\" && exit 1"

    },

    "keywords": [

        "nodejs",

        "async",

        "callbacks",

        "demonstration"

    ],

    "author": "Developer",

    "license": "MIT"

}

CODE (main.js):

const fs = require('fs');

const http = require('http');

function readUserData(userId, callback) {

    console.log(`Reading data for user ${userId}...`);

    fs.readFile(`user-${userId}.txt`, 'utf8', (err, data) => {

        if (err) {

            return callback(err);

        }

        try {

            const userData = JSON.parse(data);

            callback(null, userData);

        } catch (parseErr) {

            callback(parseErr);

        }

    });

}

function writeUserData(userId, userData, callback) {

    console.log(`Writing data for user ${userId}...`);

    fs.writeFile(`user-${userId}.txt`, JSON.stringify(userData, null, 2), (err) => {

        if (err) {

            return callback(err);

        }

        callback(null, `User ${userId} data saved successfully`);

    });

}

function fetchDataFromAPI(callback) {

    console.log('Making API request...');

    const options = {

        hostname: 'jsonplaceholder.typicode.com',

        path: '/users/1',

        method: 'GET',

        headers: {

            'Content-Type': 'application/json'

        }

    };

    const req = http.request(options, (res) => {

        let data = '';

        res.on('data', (chunk) => {

            data += chunk;

        });

        res.on('end', () => {

            try {

                const parsedData = JSON.parse(data);

                callback(null, parsedData);

            } catch (parseErr) {

                callback(parseErr);

            }

        });

    });

    req.on('error', (err) => {

        callback(err);

    });

    req.setTimeout(5000, () => {

        req.destroy();

        callback(new Error('Request timeout'));

    });

    req.end();

}

function demonstrateCallbacks() {

    console.log('=== Starting Async Callback Demonstration ===\n');

    readUserData(1, (err, userData) => {

        if (err) {

            console.error(' Read Error:', err.message);

        } else {

            console.log(' Read Success:', userData);

        }

        const newUser = {

            id: 3,

            name: 'Charlie',

            age: 28,

            email: 'charlie@example.com',

            timestamp: new Date().toISOString()

        };

        writeUserData(3, newUser, (err, result) => {

            if (err) {

                console.error(' Write Error:', err.message);

            } else {

                console.log(' Write Success:', result);

                readUserData(3, (err, verifiedData) => {

                    if (err) {

                        console.error(' Verify Error:', err.message);

                    } else {

                        console.log(' Verify Success:', verifiedData);

                    }

                });

            }

        });

    });

    setTimeout(() => {

        console.log('\n--- Testing HTTP API Call ---');

        fetchDataFromAPI((err, apiData) => {

            if (err) {

                console.error(' API Error:', err.message);

            } else {

                console.log(' API Success:');

                console.log('   User:', apiData.name);

                console.log('   Email:', apiData.email);

                console.log('   Company:', apiData.company?.name);

            }

        });

    }, 1000);

    setTimeout(() => {

        console.log('\n--- Testing Error Handling ---');

        readUserData(999, (err, data) => {

            if (err) {

                console.log(' Proper Error Handling - File not found:', err.message);

            }

        });

    }, 2000);

}

const server = http.createServer((req, res) => {

    if (req.url === '/user/1' && req.method === 'GET') {

        readUserData(1, (err, userData) => {

            if (err) {

                res.writeHead(404, { 'Content-Type': 'application/json' });

                res.end(JSON.stringify({ error: 'User not found' }));

            } else {

                res.writeHead(200, { 'Content-Type': 'application/json' });

                res.end(JSON.stringify(userData));

            }

        });

    }

    else if (req.url === '/api-data' && req.method === 'GET') {

        fetchDataFromAPI((err, apiData) => {

            if (err) {

                res.writeHead(500, { 'Content-Type': 'application/json' });

                res.end(JSON.stringify({ error: 'API call failed' }));

            } else {

                res.writeHead(200, { 'Content-Type': 'application/json' });

                res.end(JSON.stringify(apiData));

            }

        });

    }

    else {

        res.writeHead(200, { 'Content-Type': 'text/html' });

        res.end(`

            <html>

                <body>

                    <h1>Node.js Async Callback Demo</h1>

                    <p>Endpoints:</p>

                    <ul>

                        <li><a href="/user/1">/user/1</a> - Get User 1 Data</li>

                        <li><a href="/api-data">/api-data</a> - Get API Data</li>

                    </ul>

                </body>

            </html>

        `);

    }

});

const PORT = 3000;

server.listen(PORT, () => {

    console.log(`Server running at http://localhost:${PORT}`);

    demonstrateCallbacks();

});

module.exports = {

    readUserData,

    writeUserData,

    fetchDataFromAPI

};

OUTPUT:







