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(pars
eErr);
            }
        });
    });

    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)
{
                        conso
le.error(' Verify Error:',
err.message);
                    } else {
                        conso
le.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>Endpoi
nts:</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
};
```

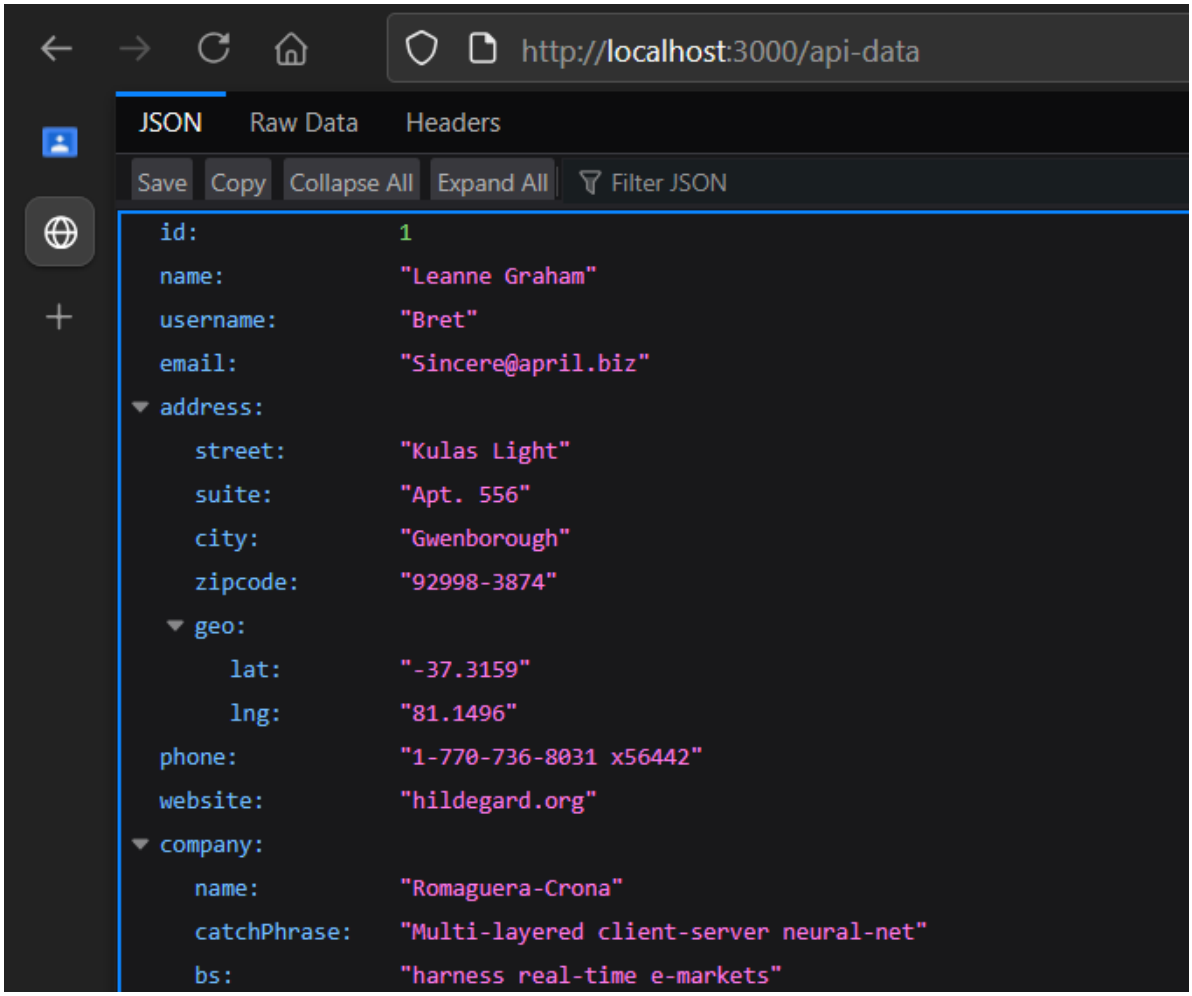
# OUTPUT:

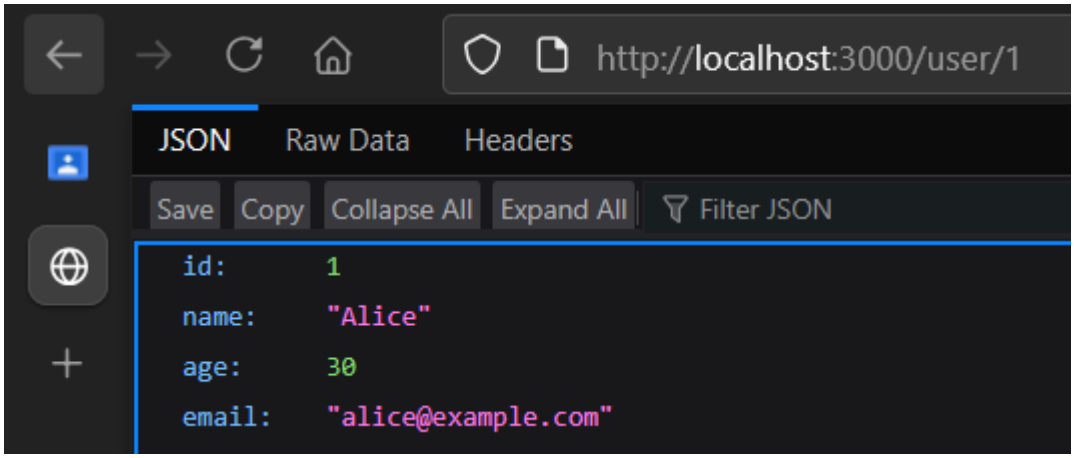
```
Abdurrahman Qureshi@DESKTOP-H2RV5MQ MINGW64 /d/Degree/SEM 5/INP/Experiments/EXP11 (master)
$ node main.js
Server running at http://localhost:3000
=== Starting Async Callback Demonstration ===

Reading data for user 1...
 Read Success: { id: 1, name: 'Alice', age: 30, email: 'alice@example.com' }
Writing data for user 3...
 Write Success: User 3 data saved successfully
Reading data for user 3...
 Verify Success: {
  id: 3,
  name: 'Charlie',
  age: 28,
  email: 'charlie@example.com',
  timestamp: '2025-10-09T12:37:24.626Z'
}

--- Testing HTTP API Call ---
Making API request...
 API Success:
   User: Leanne Graham
   Email: Sincere@april.biz
   Company: Romaguera-Crona

--- Testing Error Handling ---
Reading data for user 999...
 Proper Error Handling - File not found: ENOENT: no such file or directory, open 'D:\Degree\SEM 5\INP\Experiments\EXP11\user-999.t
xt'
```


http://localhost:3000/api-data
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
id: 1
name: "Leanne Graham"
username: "Bret"
email: "Sincere@april.biz"
address:
street: "Kulas Light"
suite: "Apt. 556"
city: "Gwenborough"
zipcode: "92998-3874"
geo:
lat: "-37.3159"
lng: "81.1496"
phone: "1-770-736-8031 x56442"
website: "hildegard.org"
company:
name: "Romaguera-Crona"
catchPhrase: "Multi-layered client-server neural-net"
bs: "harness real-time e-markets"


http://localhost:3000/user/1
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
id: 1
name: "Alice"
age: 30
email: "alice@example.com"