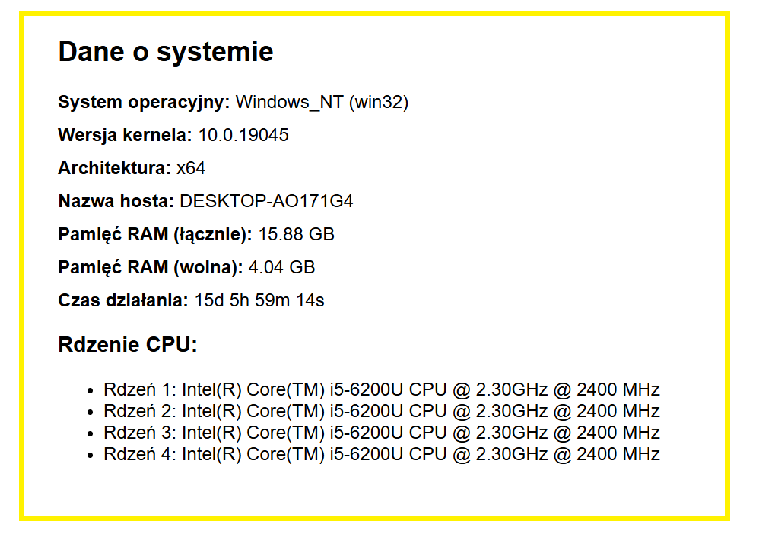
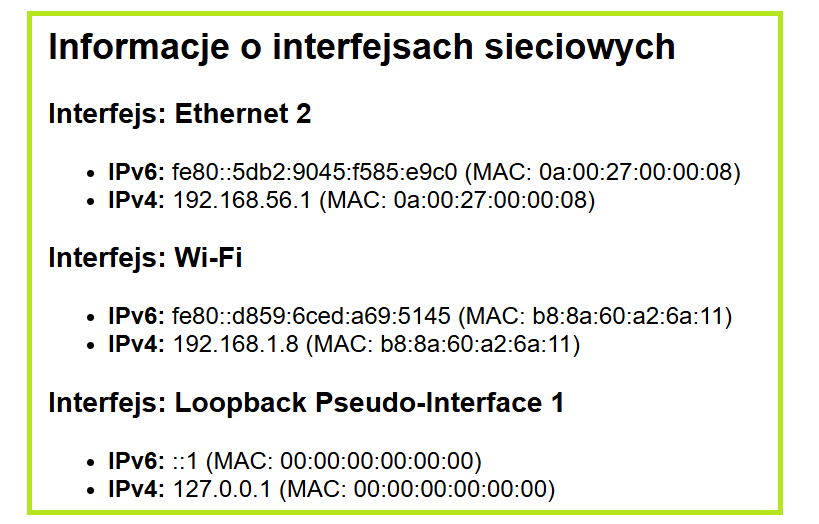
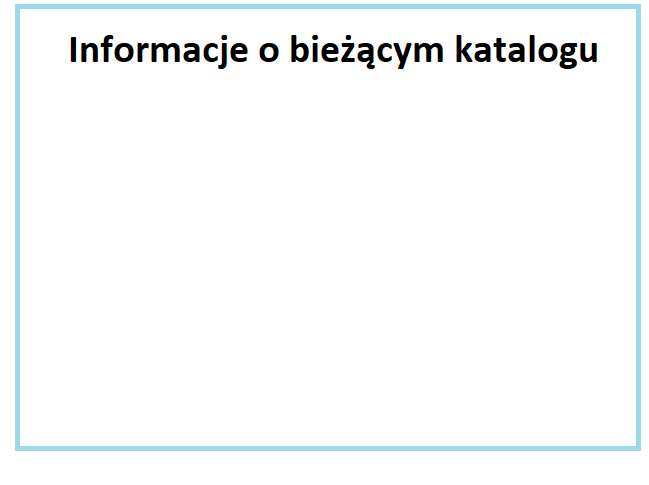
SKRYPT 26

Przygotuj konstrukcję klient-serwer. Serwer ma wystawiać 3 endpointy. W danych o systemie dodaj wyświetlanie katalogu domowego. Przygotuj endpoint – Informacje o bieżącym katalogu. Klient ma wyglądać jak poniżej:









SERWER

const express = require('express');

const cors = require('cors');

const os = require('os');

const app = express();

const PORT = 3000;

app.use(cors()); // Włącz obsługę CORS

// Endpoint: Informacje o systemie

app.get('/api/system-info', (req, res) => {

    const systemInfo = {

        osType: os.type(),

        platform: os.platform(),

        kernelVersion:

        architecture:

        hostname:

        totalMemory: (os.totalmem() / 1024 / 1024 / 1024).toFixed(2),

        freeMemory: (os.freemem() / 1024 / 1024 / 1024).toFixed(2),

        uptime: formatUptime(os.uptime()),

        cpus: os.cpus().map((cpu, index) => ({

            core: index + 1,

            model: cpu.model,

            speed: `${cpu.speed} MHz`,

        })),

    };

    res.json(systemInfo);

});

// Endpoint: Informacje o interfejsach sieciowych

app.get('/api/network-stats', (req, res) => {

    const networkInterfaces = os.networkInterfaces();

    const stats = Object.entries(networkInterfaces).map(([name, interfaces]) => ({

        interface: name,

        addresses: interfaces.map(i => ({

            family: i.family,

            address: i.address,

            mac: i.mac,

        })),

    }));

    res.json(stats);

});

// Formatowanie czasu działania systemu

function formatUptime(seconds) {

    const days = Math.floor(seconds / (24 \* 3600));

    const hours = Math.floor((seconds % (24 \* 3600)) / 3600);

    const minutes = Math.floor((seconds % 3600) / 60);

    const secs = seconds % 60;

    return `${days}d ${hours}h ${minutes}m ${secs}s`;

}

// Start serwera

app.listen(PORT, () => {

    console.log(`Serwer działa na http://localhost:${PORT}`);

});

////Endpoint podglądu katalogu//////////////

// Endpoint: Podgląd katalogu

app.get('/api/files', (req, res) => {

    const directoryPath = path.join(\_\_dirname, '/'); // Możesz zmienić na ścieżkę, którą chcesz wyświetlić

    fs.readdir(directoryPath, (err, files) => {

        if (err) {

            return res.status(500).json({ error: 'Nie udało się pobrać plików.', details: err.message });

        }

        const fileDetails = files.map(file => {

            const filePath = path.join(directoryPath, file);

            const stat = fs.statSync(filePath);

            return {

                name: file,

                isDirectory: stat.isDirectory(),

                size: stat.size,

                createdAt: stat.birthtime,

                modifiedAt: stat.mtime

            };

        });

        res.json(fileDetails);

    });

});