**Cloud Assignment S3**

# AWS BUCKET :

Code

HTML :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Cloud specs and pics</title>

</head>

<body>

    <form action="http://localhost:3000" action="GET">

        <input name="deviceid" type="text">

        <input type="submit" value="submit">

    </form>

</body>

</html>

JS File :

const http = require('http');

const AWS = require('aws-sdk');

const url = require('url');

let s3bucket = new AWS.S3({ apiVersion: '2006-03-01' });

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

    req.query = url.parse(req.url, true).query

    let params = { Bucket: 'aswanth2000cloud', Key: 'sensorDetails.json', ResponseContentType: 'application/json' };

    s3bucket.getObject(params, function(err, data) {

        if (err) {

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

            res.end("error retrieving specs.json:", err);

            console.log(err);

        } else {

            const s3json = JSON.parse(Buffer.from(data.Body).toString('utf8'));

            if (req.query.deviceid in s3json) {

                params = { Bucket: 'aswanth2000cloud', Key: 'images/' + req.query.deviceid + '.jpg', ResponseContentType: 'application/jpg' };

                s3bucket.getObject(params, function(err, data) {

                    if (err) {

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

                        res.end("error retrieving device picture:", err);

                    } else {

                        console.log("photo retrieval successful");

                        const picture = 'data:image/jpeg;base64,' + Buffer.from(data.Body).toString('base64');

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

                        res.end(`<p>${s3json[req.query.deviceid]}</p><img src="${picture}"></img>`);

                    }

                });

            } else {

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

                res.end("requested device id not found");

            }

        }

    });

});

server.listen(3000)

JSON :

{

    "sensorid1": "LM358 OP – Amp IC Small LDR 10 KΩ Resistor 10 KΩ Potentiometer White LED 220 Ω Resistor Connecting wires Power supply (9V Battery)",

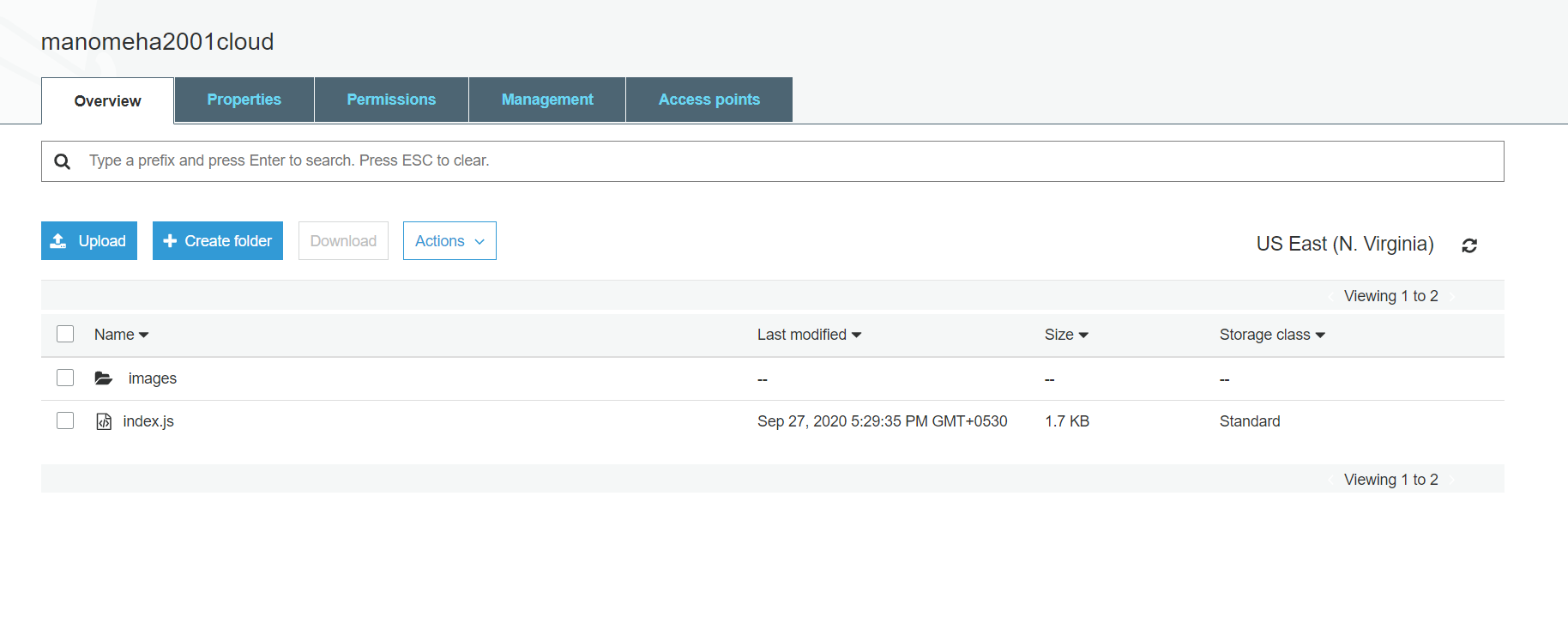
    "sensorid2": "LM359 OP – Amp IC Small LDR 9 KΩ Resistor 6 KΩ Potentiometer White LED 200 Ω Resistor Connecting wires Power supply (10V Battery)",

    "sensorid3": "LM360 OP – Amp IC Small LDR 8 KΩ Resistor 3 KΩ Potentiometer White LED 180 Ω Resistor Connecting wires Power supply (8V Battery)",

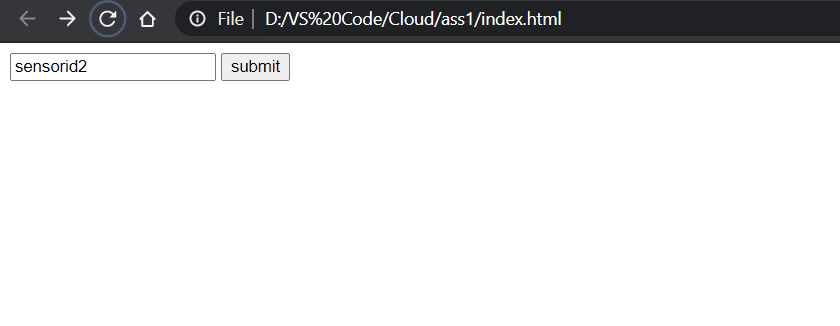
    "sensorid4": "LM362 OP – Amp IC Small LDR 6 KΩ Resistor 2 KΩ Potentiometer White LED 150 Ω Resistor Connecting wires Power supply (6V Battery)"

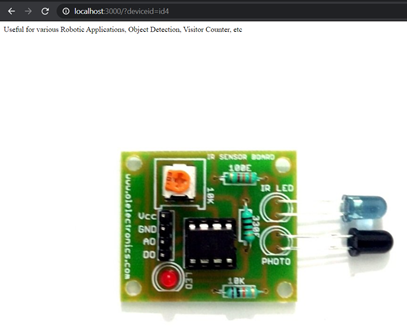
}

AWS Console 1 :

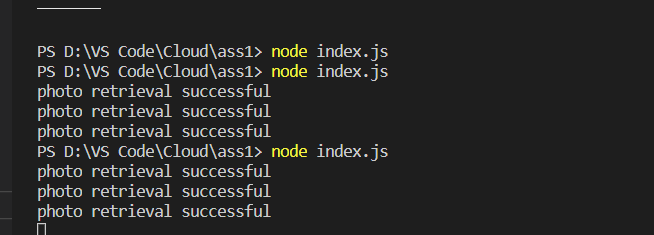


HTML OUTPUT :





Local Terminal:



# GCP Bucket :

Code :

Index1.js

const http = require('http');

const url = require('url');

const config = require('./config.json');

const { Storage } = require('@google-cloud/storage');

const { readFileSync } = require('fs');

const storage = new Storage();

async function downloadFile(filename) {

    const options = {

        // The path to which the file should be downloaded, e.g. "./file.txt"

        destination: './temp',

    };

    await storage.bucket(config.bucketName).file(filename).download(options);

    let ans = readFileSync('./temp')

    return ans

}

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

    req.query = url.parse(req.url, true).query

    downloadFile('sensorDetails.json').then((s3json) => {

        s3json = JSON.parse(s3json.toString('utf-8'))

            // const s3json = JSON.parse(Buffer.from(data.Body).toString('utf8'));

        if (req.query.deviceid in s3json) {

            downloadFile('images/' + req.query.deviceid + '.jpg').then((picture) => {

                console.log("photo retrieval successful");

                picture = 'data:image/jpeg;base64,' + picture.toString('base64');

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

                res.end(`<p>${s3json[req.query.deviceid]}</p><img src="${picture}"></img>`);

            }).catch((err) => {

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

                res.end("error retrieving device picture:", err);

            })

        } else {

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

            res.end("requested device id not found");

        }

    }).catch(err => {

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

        res.end("error retrieving specs.json:", err);

        console.log(err);

    })

});

server.listen(3000)

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Cloud specs and pics</title>

</head>

<body>

    <form action="http://localhost:3000" action="GET">

        <input name="deviceid" type="text">

        <input type="submit" value="submit">

    </form>

</body>

</html>

JSON :

{

    "sensorid1": "LM358 OP – Amp IC Small LDR 10 KΩ Resistor 10 KΩ Potentiometer White LED 220 Ω Resistor Connecting wires Power supply (9V Battery)",

    "sensorid2": "LM359 OP – Amp IC Small LDR 9 KΩ Resistor 6 KΩ Potentiometer White LED 200 Ω Resistor Connecting wires Power supply (10V Battery)",

    "sensorid3": "LM360 OP – Amp IC Small LDR 8 KΩ Resistor 3 KΩ Potentiometer White LED 180 Ω Resistor Connecting wires Power supply (8V Battery)",

    "sensorid4": "LM362 OP – Amp IC Small LDR 6 KΩ Resistor 2 KΩ Potentiometer White LED 150 Ω Resistor Connecting wires Power supply (6V Battery)"

}

Config.json :

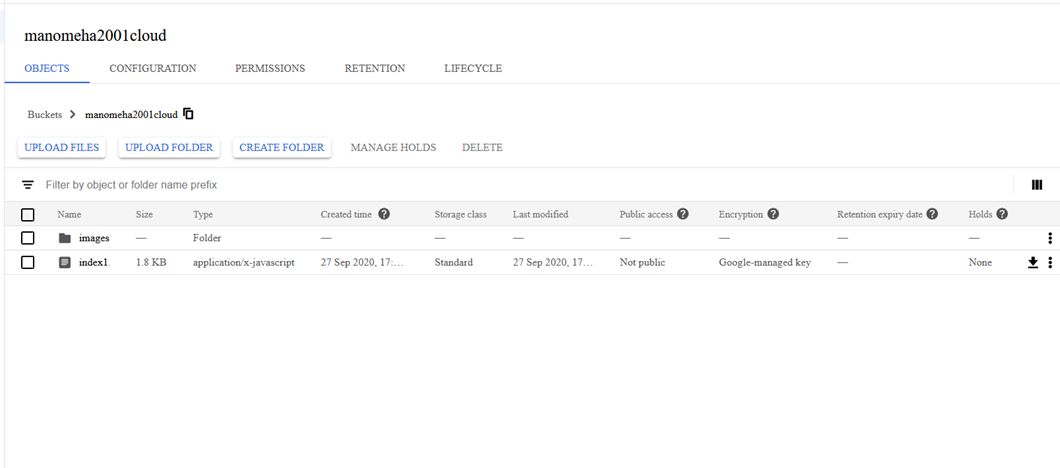
{

    "bucketName": "aswanth2000cloud"

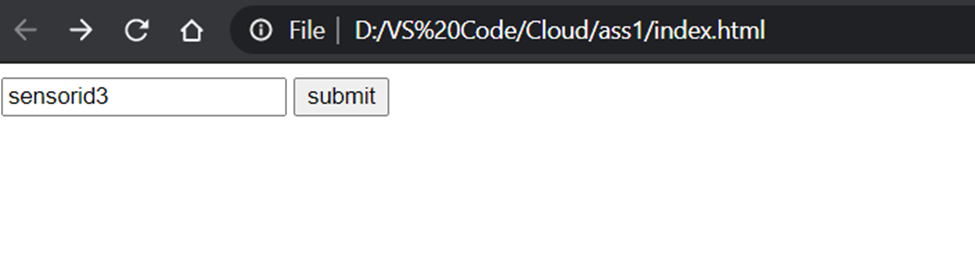
}

Output :

GCP Console:

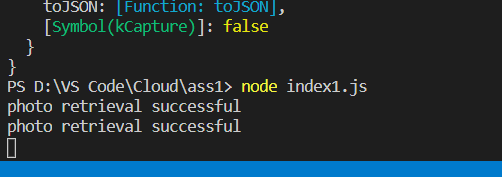


HTML Output :





Local Terminal :

 -

-----------------------------------------X-----------------------------------------