```
const express = require('express');
const app = express();

// Initialize AWS dependencies
const aws = require('aws-sdk');
aws.config.update({region: 'us-east-1'});
const s3 = new aws.S3();
const ddb = new aws.DynamoDB();
const docClient = new aws.DynamoDB.DocumentClient();

// listen to port 8008
const port = "8008";
app.listen(port, () => console.log(`Your simple cloud app is listening on port ${port}!`))

// call API when requested
app.get('/create/',createDB);
app.get('/query/:year/:title',queryDB);
app.get('/destroy/',destroyDB);

//Host the website located in the folder 'website'
app.use(express.static('public'));

// retrieve moviedata.json from s3 bucket
async function retrieveData()    {
    try {
        var params = {
            Bucket: "csu44000assign2useast20",
            Key: "moviedata.json",
           };
        const data = await s3.getObject(params).promise()
        return data.Body
    } catch (e) {
        console.log('Error',e);
    }
};

// function triggered to create database
function createDB(req, res){
    var params = {
        TableName : "Movies",
        KeySchema: [
            { AttributeName: "year", KeyType: "HASH"},
            { AttributeName: "title", KeyType: "RANGE" }
        ],
        AttributeDefinitions: [
            { AttributeName: "year", AttributeType: "N" },
            { AttributeName: "title", AttributeType: "S" }
        ],
        BillingMode: "PAY_PER_REQUEST"
        // better than provisioned throughput - in this approach all requests are added
    };
    //
    ddb.createTable(params, function (err, data) {
        if (err) {
            console.error("TABLE cannot be created due to JSON error:", JSON.stringify(err, null, 2));
```

```
        }
        else {
            console.log("TABLE created successfully with JSON:",
JSON.stringify(data, null, 2));
        }
    });

    var params = {
        TableName: 'Movies'
      };
      //  don't trigger json parse unless table is created
      ddb.waitFor('tableExists', params, function(err, data) {
        if (err) console.log(err, err.stack); // error-handling
        else  {
            retrieveData().then(result => {
                resToStr = result.toString() // convert result to string
                var moviesList = JSON.parse(resToStr);
                moviesList.forEach(function(movie) {
                    var params = {
                        TableName: "Movies",
                        Item: {
                            "year":  movie.year,
                            "title": movie.title,
                            "info":  movie.info
                        }
                    };
                    // log docClient traversal
                    docClient.put(params, function(err, data) {
                        if (err) {
                            console.error("Cannot add the entry:", movie.title, "
due to JSON error:", JSON.stringify(err, null, 2));
                        } else {
                            console.log("Title Logged:", movie.title);
                        }
                    });
                });
            })
        }
      });
}

// function triggered on query request
function queryDB(req, res)  {
    var year = parseInt(req.params.year)
    var title = req.params.title
    // additional case-sensitivity handling for title
    title = title.charAt(0).toUpperCase() + title.substring(1);

    var params = {
        TableName : "Movies",
        ProjectionExpression:"#yr, title, info.#r, info.release_date",
        KeyConditionExpression: "#yr = :yyyy and begins_with(title, :prefix)",
        ExpressionAttributeNames:{
            "#yr": "year",
            "#r": "rank"
        },
        ExpressionAttributeValues: {
            ":yyyy": year,
            ":prefix": title
```

```
        }
    };
    // query dynamodb table and send result
    docClient.query(params, function(err, data) {
        if (err) {
            console.error("QUERY cannot proceed due to error:", JSON.stringify(err,
null, 2));
        } else {
            res.json(data)
        }
    });
}

// function triggered to destroy database
function destroyDB(req, res){
    var params = {
        TableName : "Movies"
    };
    ddb.deleteTable(params, function(err, data) {
        if (err) {
            console.error("TABLE cannot be deleted due to JSON error:",
JSON.stringify(err, null, 2));
        } else {
            res = "Success"
            console.log("TABLE deleted successfully with JSON:",
JSON.stringify(data, null, 2));
        }
    });
}
```