```javascript
const dotenv = require("dotenv");
const express = require('express');
const AWS = require("aws-sdk");

// configure .env secrets
dotenv.config();


// constants
const AWS_ACCESS_KEY=process.env.AWS_ACCESS_KEY
const AWS_SECRET_KEY=process.env.AWS_SECRET_KEY
const PORT = process.env.PORT
const BUCKET="csu44000assignment220"
const FILE_LOCATION = "moviedata.json"
const TABLE_PARAMS =  {
    TableName : "Movies",
    KeySchema: [
        { AttributeName: "year", KeyType: "HASH"},  //Partition key
        { AttributeName: "title", KeyType: "RANGE" }  //Sort key
    ],
    AttributeDefinitions: [
        { AttributeName: "year", AttributeType: "N" },
        { AttributeName: "title", AttributeType: "S" }
    ],
    ProvisionedThroughput: {
        ReadCapacityUnits: 1,
        WriteCapacityUnits: 1
    }
};

const BUCKET_PARAMS = {
    Bucket: BUCKET,
    Key: FILE_LOCATION,
}

const app = express();
app.use(express.static('public')); // serves html page


// database config
AWS.config.update({
    region: "us-east-1",
    accessKeyId: AWS_ACCESS_KEY,
    secretAccessKey: AWS_SECRET_KEY
})


let dynamodb = new AWS.DynamoDB();
let s3 = new AWS.S3();


// routes
app.get('/api/create-database', async function (req, res) {

    s3.getObject(BUCKET_PARAMS, function(err,data){
        if(err){
            console.log('error:',err)
            return res.status(400).json(err);
        }

        let allMovies = JSON.parse(data.Body)

        dynamodb.createTable(TABLE_PARAMS, async function(err, data) {
            if (err) {
                console.error("Unable to create table. Error JSON:",
JSON.stringify(err, null, 2));
                return res.status(400).json(err)
            } else {
                console.log("Created table. Table description JSON:",
JSON.stringify(data, null, 2));
                await sleep(5000); // sleep so table has time to create
                var docClient = new AWS.DynamoDB.DocumentClient();
                allMovies.forEach(function (movie) {

                    var params = {
                        TableName: "Movies",
                        Item: {
                            "year":  movie.year,
                            "title": movie.title,
                            "release_date": movie.info.release_date,
                            "rank": movie.info.rank
                        }
                    };

                    docClient.put(params, function(err, data) {
                        if (err) {
                            console.error("Unable to add movie", movie.title,
". Error JSON:", JSON.stringify(err, null, 2));
                        } else {
                            console.log("PutItem succeeded:", movie.title);
                        }
                    });
                });
                return res.status(200).send("Created table and populated.")
            }
        });
    })

})

app.get('/api/get-movies', async function (req, res) {
    const {title, year} = req.query

    if(!title || !year){
        res.status(400).send('Please provide title AND year');
    }

    if(!dynamodb){
        res.status(400).send('Unable to query as table does not exist');
    }

    var docClient = new AWS.DynamoDB.DocumentClient();

    var params = {
        TableName : "Movies",
        KeyConditionExpression: "#yr = :yyyy and begins_with(title, :t)",
        ExpressionAttributeNames:{
            "#yr": "year",
        },
        ExpressionAttributeValues: {
            ":yyyy": parseInt(year),
            ":t": title
        }
    };

    docClient.query(params, function(err, data) {
        if (err) {
            console.log(err)
            return res.status(400).json(err);
        } else {
            console.log("Query succeeded.");
            var results = []
            data.Items.forEach(function(item) {
                console.log(item)
                results.push({
                    "title": item.title,
                    "year": item.year,
                    "release_date": item.release_date,
                    "rank": item.rank,
                })
            });
            return res.status(200).json(results);
        }
    });
})

app.get('/api/delete-database', async function (req, res) {

    var params = {
        TableName : "Movies"
    };

    await dynamodb.deleteTable(params, function(err, data) {
        if (err) {
            console.error("Unable to delete table. Error JSON:",
JSON.stringify(err, null, 2));
            return res.status(400).json(err)
        } else {
            return res.status(200).send('Table Deleted');
        }
    });
})

// helpers
function sleep(ms) {
    return new Promise((resolve) => {
      setTimeout(resolve, ms);
    });
}


// launch server
app.listen(PORT, function () {
    console.log(`Server is running on ${PORT} port`);
});
```