```
// CSU44000 Internet Applications Assignment 2
// Simple Movie DB App
// Pulkit Sharma


const express = require("express")
const path = require("path")
const app = express()
const PORT = 8008
const AWS = require("aws-sdk");
const AWS_ACCESS_KEY=process.env.AWS_ACCESS_KEY
const AWS_SECRET_KEY=process.env.AWS_SECRET_KEY

//Get the AWS credentials
AWS.config.update({
    region: 'us-east-1',
    accessKeyId: AWS_ACCESS_KEY,
    secretAccessKey: AWS_SECRET_KEY
});

//Check if we got the credentials
AWS.config.getCredentials(function(err) {
    if (err) console.log(err.stack);

    else {
      console.log("Access key:", AWS.config.credentials.accessKeyId);
      console.log("Secret access key:", AWS.config.credentials.secretAccessKey);
      console.log("Region: ", AWS.config.region);
      }
  });

//load the s3 movie bucket
var s3params = {
    Bucket: 'csu44000assign2useast20',
    Key: 'moviedata.json'
};

var dynamodb = new AWS.DynamoDB();
var docClient = new AWS.DynamoDB.DocumentClient();
var s3 = new AWS.S3();

//Load the HTML page
app.get("/", function (req, res) {
    res.sendFile(path.join(__dirname + "/index.html"))
});

app.listen(PORT, function () {
    console.log("AWS Movie DB running on Port: " + PORT )
});

//Create Database and table
app.post('/create', (req, res) => {
    console.log("Creating Movie Database")
    var params = {
        TableName: "Movies",
        KeySchema: [
            { AttributeName: "year", KeyType: "HASH" },
            { AttributeName: "title", KeyType: "RANGE" }
        ],
        AttributeDefinitions: [
      { AttributeName: "year", AttributeType: "N" },
      { AttributeName: "title", AttributeType: "S" }
    ],
  };

  dynamodb.createTable(params, function (err, data) {
    if (err) {
      console.error("Unable to create table. Error JSON:", JSON.stringify(err, null, 2));
    }

    else {
      console.log("Created table success. Table description JSON:", JSON.stringify(data, null, 2));
    }
  });

//Parse the movie info
  s3.getObject(s3params, function (err, data) {
    if (err) {
      console.log(err, err.stack);
    } else {
      var allMovies = JSON.parse(data.Body.toString());
       allMovies.forEach(function (movie) {
        var params = {
          TableName: "Movies",
          Item: {
            "year": movie.year,
            "title": movie.title,
            "rating": movie.info.rating,
            "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("Adding movie :", movie.title);
          }
        });
      });
    }
    console.log("Database created and populated");
  })
});

//Query the Database
app.post('/query/:title/:year', (req, res) => {
  console.log("Querying ..")
  var myArray = {
    dataEntry :[]
  }
  var year = parseInt(req.params.year)
  var title = req.params.title
  var params = {
    TableName : "Movies",
        ProjectionExpression:"#yr, title, rating, #r, #re",
        KeyConditionExpression: "#yr = :yyyy and begins_with (title, :letter1)",
        ExpressionAttributeNames:{
            "#yr": "year",
            "#r":"rank",
            "#re":"release"
        },
        ExpressionAttributeValues: {
            ":yyyy": year,
            ":letter1": title
        }
    };

//Check if query is successfull
    docClient.query(params, function(err, data) {
        if (err) {
            console.log("Unable to query. Error:", JSON.stringify(err, null, 2));
        }

        else {
            console.log("Query succeeded.");
            data.Items.forEach(function(item) {

                console.log(item.title + ' : ' + item.year + ' : ' + item.rating);
                var movieTitle = item.title
                var movieYear = item.year
                var movieRating = item.rating
                var movieRank = item.rank

                myArray.dataEntry.push(
                    {
                        Title: movieTitle,
                        Year : movieYear,
                        Rating: movieRating,
                        Rank: movieRank,
                    }
                )
            });
            res.json(myArray)
        }
    });
});

//Delete the Database
app.post('/destroy', (req, res) => {
    console.log("Destroying the Database");
    var params = { TableName : "Movies",};

    dynamodb.deleteTable(params, function(err, data) {
        if (err) {
            console.error("Unable to delete table. Error JSON:", JSON.stringify(err,
null, 2));
        }

        else {
            console.log("Deleted table. Table description JSON:",
JSON.stringify(data, null, 2));
        }
    });
});
```