```
const express = require("express");
const path = require("path");

// Load the SDK for JavaScript
var AWS = require("aws-sdk");
AWS.config.update({
  region: "eu-west-1",
  accessKeyId: "AKIAQ2JON3IN6PIOXC64",
  secretAccessKey: "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",
});

const app = express();
const port = 3000;

app.get("/", function (req, res) {
  res.sendFile(path.join(__dirname, "index.html"));
});
app.get("/createTable", createTableFunction);
app.get("/queryTable/:name/:year/:rating", queryTableFunction);
app.get("/deleteTable", deleteTableFunction);

app.listen(port, console.log(`Listening at http://localhost:${port}`));

//============================================================================

//============================================================================



function createTableFunction() {
  //Get Data From s3-------------------------------------------------------------
  var s3 = new AWS.S3();
  var dynamodb = new AWS.DynamoDB();
  var docClient = new AWS.DynamoDB.DocumentClient();
  s3.getObject(
    { Bucket: "csu44000assignment220", Key: "moviedata.json" },
    function (error, data) {
      if (error != null) {
        console.log("Failed to retrieve an object: " + error);
      } else {
        console.log("Loaded Data From Bucket");

        let myData = JSON.parse(data.Body.toString("utf-8")); //The Data

        //CreateTable in Dynamo-----------------------------------------------

        var 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: 5,
          },
        };

        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.");
        }
      });

      //FillTable in Dynamo----------------------------------------------

      let N = myData.length;
      let N2 = 500;

      for (let i = 0; i < N; i++) {
        const movie = myData[i];
        var params = {
          TableName: "Movies",
          Item: {
            year: movie.year,
            title: movie.title,
            rank: movie.info.rank,
            rating: movie.info.rating,
            image: movie.info.image_url,
            plot: movie.info.plot,
          },
        };

        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);
          }
        });
      }
    }
  }
  );
}

//============================================================================

//============================================================================

function deleteTableFunction() {
  let dynamodb = new AWS.DynamoDB();

  let params = {
    TableName: "Movies",
  };

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


//============================================================================
```

```
//============================================================================
function queryTableFunction(req, res) {
  var docClient = new AWS.DynamoDB.DocumentClient();
  console.log(req.params.year + " " + req.params.rating + " " + req.params.name)


  var params = {
    TableName : "Movies",
    ProjectionExpression:"#yr, title, rating, plot, image, #r",
    KeyConditionExpression: "#yr = :yyyy and begins_with (title, :reqName) ",
    ExpressionAttributeNames:{
        "#yr": "year",
        "#r": "rank",

    },
    ExpressionAttributeValues: {
        ":yyyy": parseInt(req.params.year),
        ":reqName": req.params.name
    }
};

  docClient.query(params, function (err, data) {
    if (err) {
      console.error("Unable to query. Error:", JSON.stringify(err, null, 2));
    } else {
      let d = []
      console.log("Query succeeded.");
      data.Items.forEach(function (item) {
        if(item.rating > req.params.rating)
        {
          d.push(item)
        }
      });
      res.send(d);
    }
  });
}
```