CSU44000 Internet Applications

Assignment#2 – A Simple Cloud Application 2021/22

Prof. Donal O'Mahony

The objective of this assignment is to write a very simple client (employing Vue.js) interacting with a server (implemented in Node.js) which in turn interacts with a Cloud-based Database (using AWS DynamoDB) and an Object stored in the Object-store (using AWS S3)

Raw data concerning movies is stored in JSON format in an object store that I have set up at :

Region: EU(Ireland)
Bucket-Name:csu44000assignment220
Object Key (FileName):moviedata.json

There is also a copy in this bucket – for any account that cannot access the EU region

Region:US East (us-east-1)
Bucket-Name:csu44000assign2useast20
Object Key (Filename): moviedata.json

You should write a simple client in Vue.js which has 3 buttons:

1. Create Database
2. Query Database with three input boxes to allow a movie name and a year and a rating to be entered
3. Destroy Database

Clicking each of these buttons will invoke API primitives on your Cloud-based server and deal with the responses.

'Create' should cause your Node.js server to make a table in a DynamoDB database – fetch the raw data from the S3 object and upload it to the newly created database. You can use a small sub-set of the fields including [title, release-date, , rating, rank]

'Query' should cause your Node.js server to find all the movies in a given year, that begin-with the entered text string with a rating higher than a given threshold.– and display them on the web-page

'Destroy' should cause the database table to be deleted.

A lot of code that you need to get this application working can be found in the AWS documentation – which shows a related example: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.NodeJs.html - feel free to copy code verbatim from this. If you get it working using the old style of API calls, you can experiment with producing a version that uses Async functions and Await.

Note: Do not set your Database ReadCapacityUnits or WriteCapacityUnits to a value greater than 1 – you won't need it and it will burn up your AWS credit allocation. Change: for the initial load of the database, your writes will exceed 1/second – try a value of 5 for this.

You can mostly ignore error handling – This exercise is mainly about the main architectural elements of the solution. Think about what you might need to do to scale this for 100 million simultaneous users.

Ideally, you will submit a .HTML file containing the client side and a single .JS file containing the server code. Both of these files should be printed-to-PDF before submission. Take a screen-shot of your working application screen in the browser – showing the Amazon Cloud URL and submit this to show how the application looks.

The submission deadline is Sunday 5th, December, at 10pm.