COM644 Full Stack Web and App Development

Week 5 Challenge

Aim: Analyse a MongoDB collection for inconsistencies and update the collection as required; add new routes and controllers to an application.

Description:

The **businessDB** database we have been working with for our sample application contains a field *review_count* for each business, as well as an array of *review* objects. However, this information is not always consistent and the *review_count* value does not always match the actual number of reviews. Your task is to identify *review_count* values that need to be changed and to perform the update.

Note:

You should work with the version of the application at the end of **Practical B3** – as this is currently the latest fully-working version of the application. This means that you are using the native MongoDB driver rather than Mongoose.

Part 1 – Create a route to check the *review_count* values for consistency

a. Add a new route **GET /checkReviews** that invokes a new controller `checkReviews()`. The new controller can be located in the **businesses.controllers.js** file.

b. Implement the new `checkReviews()` controller so that it loops across all documents in the **business** collection and generates output in the browser, such as that shown below, that identifies all documents where the number of elements in the **reviews** collection does not match the `review_count` value.

```
[{"_id":"58ac9e4ca6d33b63460d0170","review_count":6,"actual_reviews":7},
{"_id":"58ac9e4ca6d33b63460d0172","review_count":7,"actual_reviews":5},
{"_id":"58ac9e4ca6d33b63460d0174","review_count":16,"actual_reviews":18},
{"_id":"58ac9e4ca6d33b63460d0176","review_count":12,"actual_reviews":10},
{"_id":"58ac9e4ca6d33b63460d017d","review_count":4,"actual_reviews":3},
{"_id":"58ac9e4ca6d33b63460d018b","review_count":3,"actual_reviews":0},
{"_id":"58ac9e4ca6d33b63460d0177","review_count":57,"actual_reviews":51},
{"_id":"58ac9e4ca6d33b63460d0178","review_count":23,"actual_reviews":19},
{"_id":"58ac9e4ca6d33b63460d017c","review_count":119,"actual_reviews":108},
{"_id":"58ac9e4ca6d33b63460d017e","review_count":6,"actual_reviews":5},
{"_id":"58ac9e4ca6d33b63460d017f","review_count":31,"actual_reviews":29},
{"_id":"58ac9e4ca6d33b63460d0182","review_count":17,"actual_reviews":16},
{"_id":"58ac9e4ca6d33b63460d0183","review_count":63,"actual_reviews":57},
{"_id":"58ac9e4ca6d33b63460d0184","review_count":184,"actual_reviews":158},
{"_id":"58ac9e4ca6d33b63460d0185","review_count":166,"actual_reviews":158},
{"_id":"58ac9e4ca6d33b63460d0186","review_count":106,"actual_reviews":100},
{"_id":"58ac9e4ca6d33b63460d0188","review_count":7,"actual_reviews":6},
{"_id":"58ac9e4ca6d33b63460d0189","review_count":11,"actual_reviews":10},
{"_id":"58ac9e4ca6d33b63460d018a","review_count":5,"actual_reviews":4},
{"_id":"58ac9e4ca6d33b63460d018d","review_count":37,"actual_reviews":35},
{"_id":"58ac9e4ca6d33b63460d018f","review_count":4,"actual_reviews":3},
{"_id":"58ac9e4ca6d33b63460d0190","review_count":32,"actual_reviews":28},
{"_id":"58ac9e4ca6d33b63460d0193","review_count":21,"actual_reviews":17},
{"_id":"58ac9e4ca6d33b63460d019a","review_count":4,"actual_reviews":1},
```

Part 2 – Create a route to update the *review_count* value where necessary

c. Provide a second new route **GET /fixReviews** that invokes another new controller `fixReviews()`.

d. Now, implement the `fixReviews()` controller to correct the data set by updating the `review_count` value to the "real" number of reviews for every document where you previously reported a discrepancy.

Part 3 – Validate the correct operation of the new controllers

e. Prove the correct operation of your `fixReviews()` controller by re-running `checkReviews()` and confirming that it now reports an empty list.